64000

# HP64000 Logic Development System

# Model 64610S High Speed Timing/ State Analyzer Operating Manual

HEWLETT PACKARD

## CERTIFICATION

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

## WARRANTY

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

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

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

HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

### LIMITATION OF WARRANTY

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

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

### EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

## ASSISTANCE

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

*For any assistance, contact your nearest Hewlett-Packard Sales and Service Office. Addresses are provided at the back of this manual.*

FOLD HERE

---

HEWLETT PACKARD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

# BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

## HEWLETT-PACKARD

**Logic Product Support Dept.**
**Attn: Technical Publications Manager**
**Centennial Annex - D2**
**P.O. Box 617**
**Colorado Springs, Colorado 80901-0617**

---

FOLD HERE

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

# READER COMMENT SHEET

Operating Manual, Model 64610S
High Speed Timing/State Analyzer
64610-90901, October 1984

Your comments are important to us. Please answer this questionaire and return it to us. Circle the number that best describes your answer in questions 1 through 7. Thank you.

1. The information in this book is complete:

| | | |
|---|---|---|
| Doesn't cover enough (what more do you need?) | 1 2 3 4 5 | Covers everything |

2. The information in this book is accurate:

| | | |
|---|---|---|
| Too many errors | 1 2 3 4 5 | Exactly right |

3. The information in this book is easy to find:

| | | |
|---|---|---|
| I can't find things I need | 1 2 3 4 5 | I can find info quickly |

4. The Index and Table of Contents are useful:

| | | |
|---|---|---|
| Helpful | 1 2 3 4 5 | Missing or inadequate |

5. What about the "how-to" procedures and examples:

| | | |
|---|---|---|
| No help | 1 2 3 4 5 | Very helpful |
| Too many now | 1 2 3 4 5 | I'd like more |

6. What about the writing style:

| | | |
|---|---|---|
| Confusing | 1 2 3 4 5 | Clear |

7. What about organization of the book:

| | | |
|---|---|---|
| Poor order | 1 2 3 4 5 | Good order |

8. What about the size of the book:

| | | |
|---|---|---|
| too big/small | 1 2 3 4 5 | Right size |

Comments: _____

_____

_____

_____

Particular pages with errors?

_____

Name (optional): _____
Job title: _____
Company: _____
Address: _____

**Note:** If mailed outside U.S.A., place card in envelope. Use address shown on other side of this card.

FOLD HERE

---

HEWLETT PACKARD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

# BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE


**HEWLETT-PACKARD**
Logic Product Support Dept.
Attn: Technical Publications Manager
Centennial Annex - D2
P.O. Box 617
Colorado Springs, Colorado 80901-0617


---

FOLD HERE

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

# READER COMMENT SHEET

Operating Manual, Model 64610S
High Speed Timing/State Analyzer
64610-90901, October 1984

Your comments are important to us. Please answer this questionaire and return it to us. Circle the number that best describes your answer in questions 1 through 7. Thank you.

1. The information in this book is complete:

| | | |
|---|---|---|
| Doesn't cover enough (what more do you need?) | 1 2 3 4 5 | Covers everything |

2. The information in this book is accurate:

| | | |
|---|---|---|
| Too many errors | 1 2 3 4 5 | Exactly right |

3. The information in this book is easy to find:

| | | |
|---|---|---|
| I can't find things I need | 1 2 3 4 5 | I can find info quickly |

4. The Index and Table of Contents are useful:

| | | |
|---|---|---|
| Helpful | 1 2 3 4 5 | Missing or inadequate |

5. What about the "how-to" procedures and examples:

| | | |
|---|---|---|
| No help | 1 2 3 4 5 | Very helpful |
| Too many now | 1 2 3 4 5 | I'd like more |

6. What about the writing style:

| | | |
|---|---|---|
| Confusing | 1 2 3 4 5 | Clear |

7. What about organization of the book:

| | | |
|---|---|---|
| Poor order | 1 2 3 4 5 | Good order |

8. What about the size of the book:

| | | |
|---|---|---|
| too big/small | 1 2 3 4 5 | Right size |

Comments: _____

_____

_____

_____

Particular pages with errors?

_____

Name (optional): _____
Job title: _____
Company: _____
Address: _____

**Note:** If mailed outside U.S.A., place card in envelope. Use address shown on other side of this card.

## SAFETY SUMMARY

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

**GROUND THE INSTRUMENT.**

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

**DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE.**

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

**KEEP AWAY FROM LIVE CIRCUITS.**

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

**DO NOT SERVICE OR ADJUST ALONE.**

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

**DO NOT SUBSTITUTE PARTS OR MODIFY INSTRUMENT.**

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

**DANGEROUS PROCEDURE WARNINGS.**

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

WARNING

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

## SAFETY SYMBOLS

**General Definitions of Safety Symbols Used on Equipment or in Manuals.**

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

Indicates dangerous voltage (terminals fed from the interior by voltage exceeding 1000 volts must be so marked).

 **OR** 

Protective conductor terminal. For protection against electrical shock in case of a fault. Used with field wiring terminals to indicate the terminal which must be connected to ground before operating equipment.

Low-noise or noiseless, clean ground (earth) terminal. Used for a signal common, as well as providing protection against electrical shock in case of a fault. A terminal marked with this symbol must be connected to ground in the manner described in the installation (operating) manual, and before operating the equipment.

 **OR** 

Frame or chassis terminal. A connection to the frame (chassis) of the equipment which normally includes all exposed metal structures.

Alternating current (power line).



Direct current (power line).

Alternating or direct current (power line).


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


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

**NOTE:**

The NOTE sign denotes important information. It calls attention to procedure, practice, condition or the like, which is essential to highlight.

HEWLETT PACKARD

OPERATING MANUAL

# MODEL 64610S
# HIGH SPEED TIMING/STATE ANALYZER



Manual Part No. 64610-90901

PRINTED: October 1984

## PRINTING HISTORY

Each new edition of this manual incorporates all material updated since the previous edition. Manual change sheets are issued between editions, allowing you to correct or insert information in the current edition.

The print date changes only when each new edition is published. Minor corrections or additions may be made as the manual is reprinted between editions. Vertical bars in a page margin indicate the location of reprint corrections.

**First Edition . . . . . October 1984 64610-90901**

## 64600S TIMING ANALYZER OBSOLETE

The HP64610S High Speed Timing/State Analyzer obsoletes the HP64600S Timing Analyzer. New software exists and replaces the HP64600 software. The information in this manual applies directly to the 64600S Timing Analyzer, however it also contains information on the new State Analysis enhancement and the use of 24 and 32 channels of timing. For more information on the State Analysis capabilities, see Chapters 1 and 2.

## SOFTWARE VERSION NUMBERS

This operating manual applies to software version number 64610-1.00. Your HP64000 software is identified with this software version number.

## DUPLICATING SOFTWARE

Before using the flexible disc(s) provided with this product, make a work copy. Retain the original disc(s) as the master copy and use the work copy for daily use. The procedure for duplicating the master flexible disc(s) is included in Chapter 2.

Specific rights to use one copy of the software product(s) are granted for use on a single, stand-alone development station or a cluster of development stations which boot from a single mass storage device.

Should your master-copy become damaged, replacement discs are available through your Hewlett-Packard sales and service office.

# Table Of Contents

# Table Of Contents (Cont'd)

# Table Of Contents (Cont'd)

## Chapter 6 - INTERACTING WITH OTHER ANALYZERS (Cont'd)



# List Of Illustrations



# List Of Tables

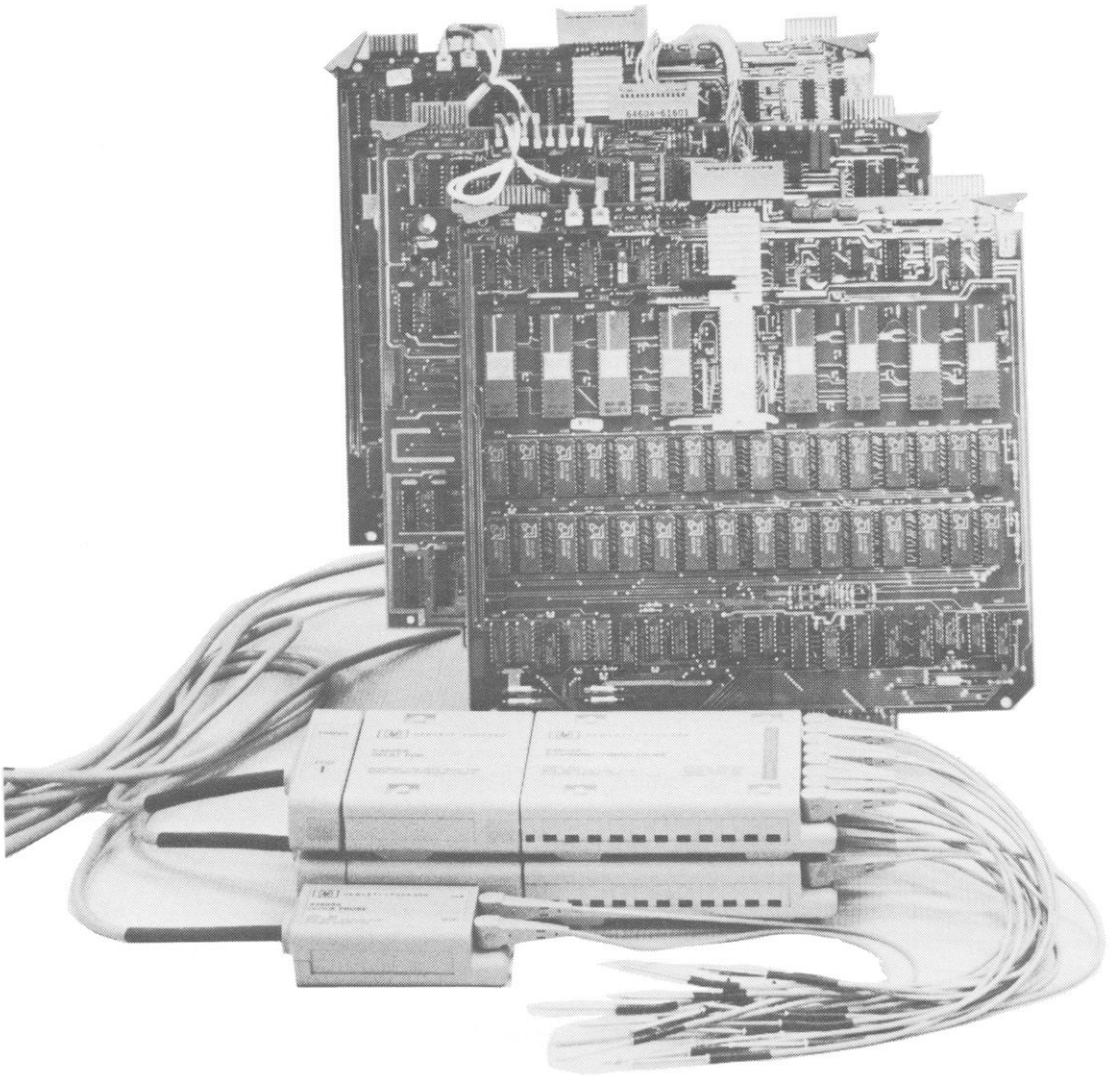


**Figure 1-1. High Speed Timing/State Analyzer (16 channels)**

# Chapter 1

# GENERAL INFORMATION

## INTRODUCTION

This manual describes how to operate the Model 64610S High Speed Timing/State Analyzer when installed in the Hewlett-Packard 64000 series Logic Development System. The content of this manual is described in this chapter.

## MANUAL CONTENT

**Chapter 1 General Information** explains the content of this manual, provides a description of the High Speed Timing/State Analyzer, and lists the features of the analyzer. Software Materials Subscription information is included here.

**Chapter 2 Installation** provides an inventory list of parts that make up the analyzer. The installation procedure is located here along with illustrations. Other information included in this chapter consists of installing and duplicating software and verifying analyzer performance.

**Chapter 3 Getting Started** contains information on setting up the analyzer, accessing the analyzer, using State Analysis and Timing Analysis, and provides descriptions of the measurement modes. It also shows you how to define labels, how to use the specification menus and gives you an example measurement to perform. Entering numeric values is also explained.

**Chapter 4 Configuring The Analyzer** shows you how to configure the analyzer using various measurement setups. It also contains information on performing default measurements.

**Chapter 5 Examples Of Analyzer Use** contains examples organized by description and command line listings. This information will enable you to perform specific examples that demonstrate the features of the High Speed Timing/State Analyzer.

**Chapter 6 Interacting With Other Analyzers** explains the interactions when working with more than one analyzer in an HP64000 development station.

**Appendix A Softkey Glossary** lists softkeys and their descriptions.

**Appendix B Analyzer Syntax Diagrams** includes command diagrams for performing tasks.

**Appendix C Storing Measurement Data In Binary** explains the measurement data file.

**Appendix D Performing Accurate Time Interval measurements** describes how to make accurate time interval measurements.

**Appendix E Analyzer Specifications** lists operating specifications for the analyzer.

**Appendix F Theory Of Operation** explains the analyzer hardware and software.

**Appendix G Analyzer Error And Status Messages** lists messages that you may encounter.

## HIGH SPEED TIMING/STATE ANALYZER FEATURES

Features of the High Speed Timing/State Analyzer are outlined below.

* 200/400 MHz timing (asynchronous) analysis.
* External clock rate to 125 MHz for state (synchronous) analysis.
* One external clock and one state clock qualifier.
* Maximum set-up time of 4 nS and hold time of 0.5 nS.
* Expandable to 32 channels of asynchronous or synchronous analysis.
* Pattern duration and transition triggering.
* 4k memory depth (8k at 400 MHz).
* Glitch Capture and Dual Threshold asynchronous measurement modes.
* User Defined Labels.
* Search for specified events in data acquisition memory.
* Automatic marking of specified events in data acquisition memory.
* Calculating statistics on marked events.
* Using marked events to qualify execution run.
* Processing trace list data to show specified patterns or marks.
* Processing asynchronous trace list data into pseudosynchronous state listings.
* Storing measurement data along with the system configuration.
* Comparing stored and current measurements.
* Interactive Measurements with other analysis modules.

Refer to Chapter 5 for examples of analyzer use.

## ANALYZER DESCRIPTION

The High Speed Timing/State Analyzer is a software and hardware troubleshooting instrument that runs asynchronous analysis (timing) on 32 channels at 200 MHz (16 channels at 400 MHz) or runs synchronous analysis (state) on 32 channels up to 125 MHz. The triggering modes allow precise positioning of the display window to locate timing margin and interaction problems.

The High Speed Timing/State Analyzer may be configured as an 8-, 16-, 24-, or 32-channel subsystem (see table 2-1). The analysis subsystem consists of a Control Board, a Clock Probe, up to four Acquisition Boards, up to four Data Probes, and up to four Data Delay Modules.

**NOTE**

**The Clock Probe and the Data Delay Module(s) are used for State Analysis only.**

The High Speed Timing/State analyzer subsystem is installed in any HP 64000 development station (for 64100A development stations with serial prefixes below 2149A, contact your HP representative). Multiple subsystems can operate in a single HP 64100A development station.

## WHAT IS SMS?

Hewlett-Packard offers a Software Materials Subscription (SMS) to provide timely and comprehensive information for the users of the Model 64000 Logic Development System. This service can help maximize the productivity of your HP system by insuring that the latest product enhancements and software revisions are being utilized. This service also guarantees that your system is fully compatible with new products that are introduced.

By purchasing SMS, you will obtain the following:

Software Updates
Reference Manual Updates
Software Problem Reporting
Software Status Bulletins
General User Information

**Software Updates** may address specific anomalies in HP software or enhance the capability of the HP software in your system.

**Reference Manual Updates** assure that you always have the most recent documentation on a timely basis, and are aware of how to use any new features on the latest software releases.

**Software Problem Reporting** is provided so that you may inform HP of a discrepancy or problem found in the HP 64000 software or documentation.

**Software Status Bulletins** contain timely information on the reported operational status of HP software and documentation. These bulletins also provide temporary corrections or ways to work around anomalies in HP software which have been located by HP personnel or HP 64000 users. You may reference these bulletins to see if a solution is already documented.

**General User Information** is documentation that contains operational tips, programming techniques, application notes, latest listings of software products and reference manuals, and other items of general interest to HP 64000 users.

Consult with your local HP Field Representative for a complete list of available software update products (64XXXAU), one-time product updates (64XXXAX), and current prices.

# Chapter 2

# INSTALLATION

## INTRODUCTION

This chapter contains information on:

* What Makes Up The Analyzer?
* Analyzer Options
* Software Modules Needed To Operate The Analyzer
* Checking The Contents Received
* Installing The Analyzer
* Setting The Development Station Address Switches
* Installing The Analyzer Components
* Analyzer Board Configurations
* Installing and Duplicating Software
* Verifying Analyzer Performance

## WHAT MAKES UP THE ANALYZER?

The High Speed Timing/State Analyzer consists of the hardware and software components listed below.

**Hardware:**

Model 64601B Timing/State Control Board
Model 64602A 8-Channel Acquisition Board
Model 64604A Probe
* Model 64605A Clock Probe
* Model 64606A Data Delay Module

Model 64963A Timing Bus

**Software:**

High Speed Timing/State

* These components are used for State Analysis only.

**Table 2-1. High Speed Timing/State Analyzer Options**

| Analyzer Components | 8 CHANNELS (64610S) | 16 CHANNELS (OPT 016) | 24 CHANNELS (OPT 024) | 32 CHANNELS (OPT 032) |
|---|---|---|---|---|
| 64601B Control | 1 | 1 | 1 | 1 |
| 64602A Acquisition | 1 | 2 | 3 | 4 |
| 64604A Probe | 1 | 2 | 3 | 4 |
| * 64605A Clock Probe | 1 | 1 | 1 | 1 |
| * 64606A Data Delay | 1 | 2 | 3 | 4 |
| 64963A Timing Bus | 1 (2 position) | opt 001 (3 position) | opt 002 (4 position) | opt 003 (5 position) |

* These components are used for State Analysis only.

## SOFTWARE MODULES NEEDED TO OPERATE THE ANALYZER

The software modules that you must have in order to operate the High Speed Timing/State Analyzer are listed below. (To observe the modules on your flexible disc, press ***BACKUP floppy sys_gen*** (RETURN). Once in the system generator, press ***show local <DISC #>*** (RETURN).)

System Modules:
- FLOPPY_OP_SYS
- MEAS_SYS
- TIMING

Peformance Verification Modules:
- OPTION_TEST
- PV_TIMING

## CHECKING THE CONTENTS RECEIVED

Verify that you have received all of the hardware, software and manuals for your analyzer subsystem before you begin installing the analyzer (see table 2-1). If you have not received all the components you requested, or if any of the components appear to be damaged, contact your Hewlett-Packard representative.

# INSTALLING THE ANALYZER

To install the High Speed Timing/State Analyzer, complete the steps listed below. **The High Speed Timing/State Analyzer should be installed by qualified personnel only.**

1. Set the development station address switches.
2. Install the analyzer components.
3. Boot or install software.

CAUTION

**Turn off power to the HP 64000 before installing the analyzer.**

## Setting The Development Station Address Switches

**NOTE**

Most development stations contain seven switches on the rear panel, however, some earlier models contain only five. The following instructions show switch settings for the earlier models in parentheses.

The address switches are a set of seven (or five) switches on the HP 64000 rear panel. Refer to the label located beside the address switches and set these switches as described below.

1. If you are operating as **part of a system** (with hard disc), set switches one through five (or one through three) to select the system address for your development station. Set switches six and seven (or four and five) to select SYSTEM BUS operation.

2. If you are operating as a **stand-alone instrument** (with flexible disc only), switches one through five (or one through three) are not used. In this case, set switches six and seven (or four and five) to one of the local mass storage (LMS) positions.

## Installing The Analyzer Components

The High Speed Timing/State Analyzer consists of a Control Board, Acquisition Board(s), Probe(s), Clock Probe and Data Delay Module(s). The following procedure gives you a step-by-step approach for installing all the components of the High Speed Timing/State Analyzer. Refer to Table 2-1 for information on the components needed for various configurations.

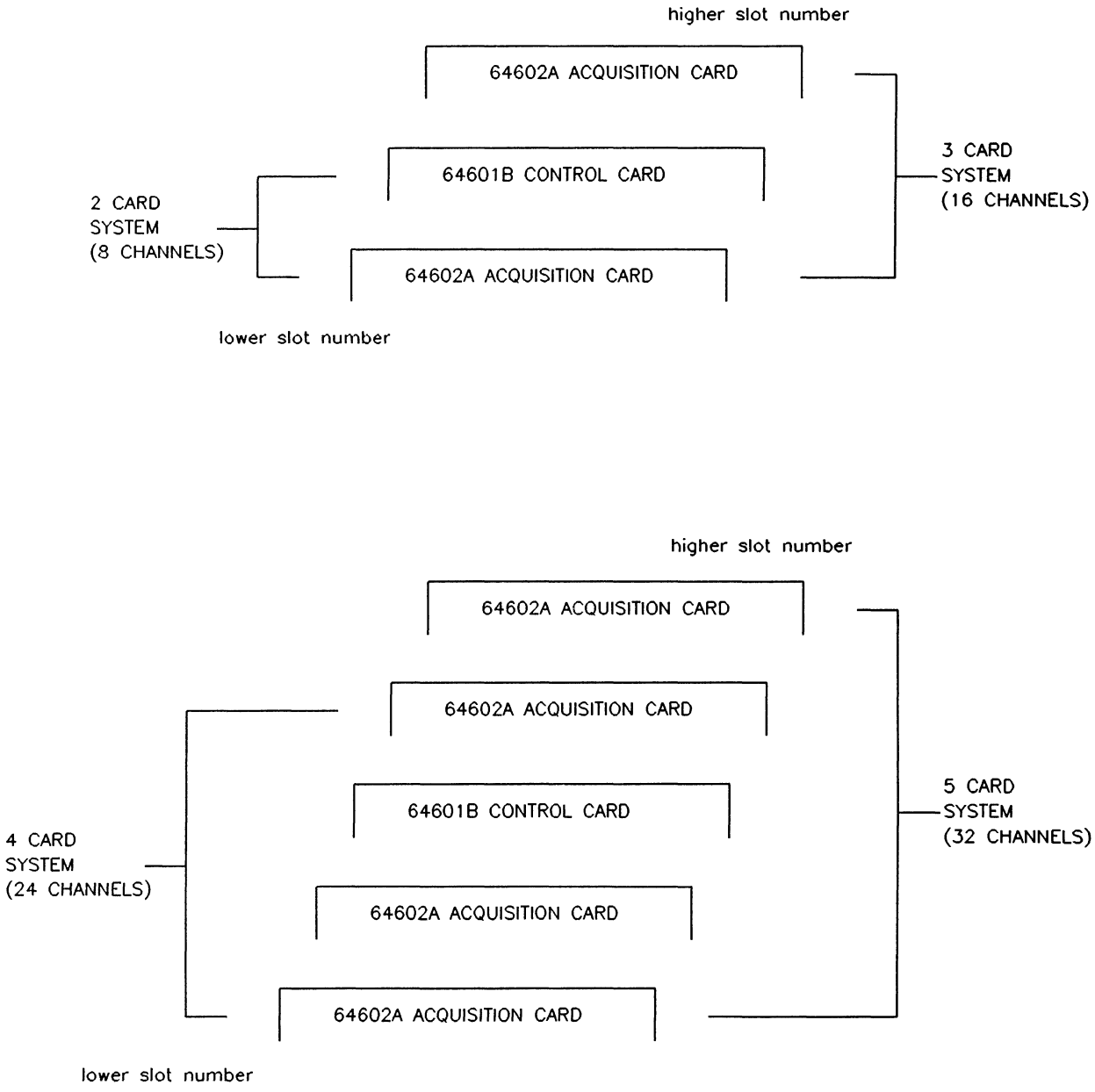



**Figure 2-1. Analyzer Board Configurations**

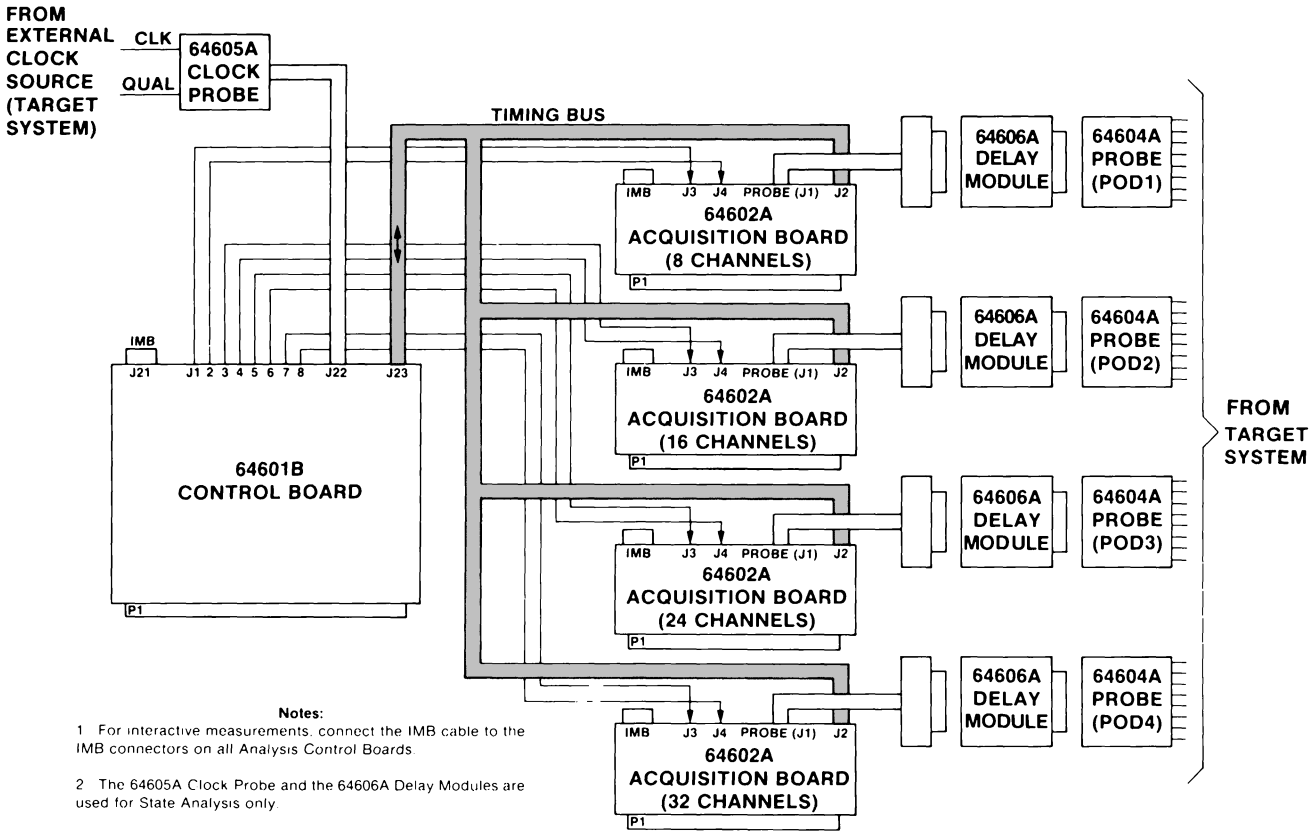



**Figure 2-2. High Speed Timing/State Analyzer Installation**

## Installing The Analyzer Components (Cont'd)

An 8-channel subsystem uses 2 boards (Control and Acquisition). Three boards are used in a 16-channel sybsystem, four boards in a 24-channel subsystem and five boards in a 32-channel subsystem.

**8-channel subsystem** - the Acquisition Board will be installed in the lower slot number of the development station. The Control Board will be installed next to the Acquisition Board in the next higher slot number.

**16-channel subsystem** - one Acquisition Board will be installed in the slot number lower than the Control Board, and the other Acquisition Board will be installed in the slot number higher than the Control Board.

**24-channel subsystem** - two Acquisition Boards will be installed in the slot numbers lower than the Control Board and the other Acquisition Board will be installed in the slot number higher than the Control Board.

**32-channel subsystem** - two Acquisition Boards will be installed in the slots lower than the Control Board and the other two Acquisition Boards will be installed in the slots higher than the Control Board.

The word ***time_st*** will appear in the monitor display when one timing/state analysis subsystem (with software) is installed. If more than one is installed, the number of the slot containing the Control Board will appear on the softkey label line in the measurement systems display beside the word ***time_st*** (i.e.: ***time_st_4***). This number identifies the analyzer with a slot number so that two analyzers can be installed in the same HP 64000 with the unique identities of each analyzer maintained. If you do not have **State Analysis** capability, the word ***timing*** will appear in place of ***time_st***. **Install the analyzer components as described below.**

1. Determine the number of channels to be used.

2. For any of the configurations listed above, install boards in the lower slot number first, then install the boards in the higher development station slot numbers. To actually install a board in the development station, hold the board by the extractor levers at the top and insert into a card slot with the components on the board facing the front of the development station. Push down until the connector of the board seated firmly.

3. Connect the coaxial clock cables from the Control Board (CB) to the Acquisition (ACQ) Board(s). These cables are supplied with each Acquisition Board and must be connected in pairs. Connect these cables as described below.

8-channels - CB connectors J1 and J2 to ACQ1 connectors J3 and J4.
16 channels - CB connectors J3 and J4 to ACQ2 connectors J3 and J4.
24 channels - CB connectors J5 and J6 to ACQ3 connectors J3 and J4.
32 channels - CB connectors J7 and J8 to ACQ4 connectors J3 and J4.

4. Connect the timing bus to J23 on the Control Board and to J2 on the Acquisition Board(s). This connection is on the right side of the boards when viewed from the front of the development station.

## Installing The Analyzer Components (Cont'd)

5. Connect the Model 64604A Probe to J1 on each Acquisition Board. The cable connecting the probe to the Acquisition Board is supplied with the Acquisition Board. Make sure you align pin 1 on the probe cable with pin 1 on the Acquisition Board connector (left side of the connector J1 labeled PROBE).

6. If you want to set up **State Analysis**, connect the Model 64605A Clock Probe to J22 on the Control Board. Make sure to align pin 1 on the cable and the connector.

7. For **State Analysis**, you must also connect the data delay module(s) between the end of the probe cable(s) and the probe(s). **Remove the data delay module(s) if you are running timing analysis only.**

8. If you are performing interactive measurements (triggering other analyzers), connect the IMB cable to the IMB connectors on the Control and Acquisition Boards (see figure 2-2) and to the other analyzer control boards included in the measurement.

**NOTE**

**Insert the metal shield of the probe cables into the cable clamps at the top rear of the development station to shield radio frequency interference.**

## Installing Software

**NOTE**

Before using the flexible disc(s) provided with this product, make a work-copy. Retain the original disc(s) as the master-copy and use the work copy for daily use. The procedure for duplicating the master flexible disc(s) is included in this chapter.

Specific rights to use one copy of the software product(s) are granted for use on a single, stand-alone development station or a cluster of development stations which boot from a single mass storage device.

If your master-copy becomes damaged, replacement discs are available through your Hewlett-Packard sales and service office.

To install your software onto the hard disc of a cluster, follow the steps outlined below.

1. Make a work-copy flexible disc as described in this chapter.
2. Insert the work-copy flexible disc in disc drive #0.
3. Copy the desired modules to the hard disc in your cluster.
4. Remove the work-copy flexible disc from disc drive #0.

In stand-alone mode, the flexible disc remains in the HP 64000.

## Duplicating Software

The flexible discs you received with your High Speed Timing/State Analyzer contain the software you will use. You should make a copy of this software for your use and store the originals that you received. The procedure outlined below shows you how to make a duplicate set of your master flexible discs.

1. Turn on power to the development station. The development station will place the monitor level of softkeys on the screen.

2. Press the following softkeys:

*etc* *etc* *<BACKUP>* *floppy* *utilities* RETURN

The screen will show a description of the floppy utilities routines.

CAUTION

**Do not write on your flexible disc with a ball point pen.**
**Doing this can cause damage to the floppy.**

3. Obtain a new blank flexible disc. Using stick-on labels or a felt tip pen, identify the flexible disc with the name of the module you intend to store on it.

4. Install the master flexible disc in disc drive 0 of your development station.

5. Install the new disc with your label in disc drive 1.

6. Press the following softkeys:

*format* **1** RETURN

7. When the formatting of disc 1 is complete, press the following keys:

*duplicate* **0** RETURN

8. The system will copy the software on disc 0 to disc 1. When the copy is complete, open both disc drives and remove the discs.

9. Repeat steps 3 through 8 to copy all of the master flexible discs. When you are finished, store the master flexible discs in a safe place.

## Duplicating Software For Stand-Alone Operation

If you are operating a stand-alone terminal, you can make one or two flexible discs containing just the modules you use. Doing this can minimize the times you have to remove and replace discs during operation.

For **stand-alone operation**, the modules to use are: FLOPPY_OP_SYS, MEAS_SYS and TIMING.

For **stand-alone performance verification**, the modules to use are: FLOPPY_OP_SYS, OPTION_TEST and PV_TIMING.

1. Install a new disc in disc drive 1. Format this disc by pressing the following keys:

*format* **1** RETURN

2. Install your master disc with the SYS_GEN module in disc drive 0 and the new formatted disc in disc drive 1.

3. Press the following keys:

*end* RETURN

*floppy sys_gen* RETURN

*show local* **0** *and local* **1** RETURN

The screen will show a list of the software modules on your user flexible disc under disc #0.

4. For each module you want on your user-defined disc, press the following keys:

*copy* **<module number or all>**

*from local* **0** *to local* **1** RETURN

5. After you have copied all of the desired modules, open disc drive 0 and remove the master flexible disc. Store it in a safe place.

6. To copy modules from another master flexible disc, insert the master disc in disc drive 0 and press the following keys:

*show local* **0** *and local* **1** RETURN

7. Repeat steps 3 through 6 to load all the desired modules on your user-defined disc in drive 1. If your user-defined disc becomes full, remove it from disc drive 1 and insert another formatted disc. Continue to load modules from discs inserted in drive 0 to the user-defined disc in drive 1. When you have loaded all of the desired modules on your user-defined disc, press the following keys:

*end* RETURN

## Duplicating Software For Stand-Alone Operation (Cont'd)

8. You can make a duplicate set of user-defined discs (via floppy utilities) and keep them with the master flexible discs. Doing this will provide you with a master set of user-defined flexible discs that can save you time in making future duplicates.

**NOTE**

**Do not write-protect the user-defined discs used in the stand-alone system. If you write-protect your user-defined when working in the stand-alone mode, you will not be able to write files (such as trace and data files) to that disc.**

## VERIFYING ANALYZER PERFORMANCE

Option test is provided for verifying proper operation of the High Speed Timing/State Analyzer. To run option test, follow the steps outlined below and observe the pass or fail results on the screen as the test completes.

**NOTE**

**All probes must be disconnected from signal sources to obtain proper performance verification results.**

1. Press *opt_test* (RETURN)
2. Press *<slot #>* . Enter the slot # of the Analyzer Control Board. (RETURN)
3. Press *run* (RETURN)
4. Observe the screen for a pass or fail status.

If PASS appears on the screen, the analyzer is set up correctly and is operating properly. If FAIL appears on the screen, a problem exists with the analyzer configuration or with the functionality of one or more of the analyzer components. In case of a FAIL status, check the analyzer setup (see Chapter 2 - Installation). If the problem persists, refer to the High Speed Timing/State Analyzer Service Manual or call your Hewlett-Packard representative.

# Chapter 3

# GETTING STARTED

## INTRODUCTION

This chapter is designed to familarize you with the High Speed Timing/State Analyzer. In this chapter you will learn how to:

* Set up the High Speed Timing/State Analyzer
* Access the Analyzer
* Perform an Example Measurement
* Choose a Measurement Mode
* Use Timing Analysis
* Use State Analysis
* Define Signal Labels
* Use the Specification Menus
* Enter Numeric Values

## SETTING UP THE ANALYZER

In setting up the analyzer to perform a specific measurement, you will first need to determine the signals in the unit under test that you want to analyze. You should also assign yourself a "user id" for convenience in keeping track of your files.

### Assigning A Userid

Assign yourself a user identification word by following these steps:

1. Press ***etc*** until ***userid*** appears
2. Press ***userid***
3. Type in any word beginning with a letter (all letters must be capitalized).
   *(The development station will accept up to 6 characters.)*
4. Press RETURN

Now any work you do will be accomplished under your userid. You can create additional userid's for the various types of work that you will be doing.

### Determining Signals To Analyze

Choose signals in the system under test that you want to either analyzer or observe. You may want to look at signals such as: clock, read/write, status, address, enables, etc. Your knowledge of the system under test and the existing problem will help you decide which signals to analyze. Once you have determined which signals you want to analyze, connect your probe leads to those signals.

## ACCESSING THE ANALYZER

The first softkey label line at the system monitor level will contain either a *time_st* (*timing* if no State Analysis capability is present) or *meas_sys* softkey. If the High Speed Timing/State Analyzer is the only module installed in the development station, *time_st* will appear in the list of softkeys. If other subsystems (such as emulation) are present in the development station, *meas_sys* will appear.

Press *time_st* or *meas_sys* , as appropriate. Press RETURN.

If you pressed:

*time_st* - the High Speed Timing/State Analyzer is accessed and the default trace specification appears on the screen.

*meas_sys* - the timing/state softkey appears on the screen, along with the names of other modules in the HP 64000. (This is the measurement system level of softkeys which allows access to any one of the modules installed in the development station.) Press *time_st* and RETURN to gain access to the analyzer. The default trace specification now appears on the screen.

**NOTE**

If there is more than one analysis module installed in the HP 64000, the timing/state softkey will be followed by a number *(time_st_4)*. The number indicates the slot number in the development station where the analysis control board is located.

## PERFORMING AN EXAMPLE MEASUREMENT

This example is provided to help you become more familar with the High Speed Timing/State Analyzer. In this example you will use an 8-channel timing analysis system to:

1. Connect probe leads to the signals under test
2. Run the Activity Test
3. Define a measurement mode
4. Define labels
5. Set up trigger specifications
6. Execute the measurement
7. Display the defined labels
8. Observe the trace list specification

### 1. Connect probe leads to the signals under test.

a. Disconnect power to the system under test.

b. Locate the most accessible area to probe the signals you have chosen.

c. Begin connecting the probe leads (POD1) to the signals under test, starting with bit 0. Using an 8085 microprocessor in the example target system (the system under test), refer to the table below to observe eight example signals.

**Table 3-1. Example Signals To Analyze**

| Probe Bit | 8085 Microprocessor Signal | Pin |
|---|---|---|
| 0 | CLK | 37 |
| 1 | LRD | 32 |
| 2 | LWR | 31 |
| 3 | ALE | 30 |
| 4 | S0 | 29 |
| 5 | S1 | 33 |
| 6 | INTR | 10 |
| 7 | INTA | 11 |

d. Turn on power to the system under test.

## 2. Run the Activity Test.

a. Press ***show format*** (RETURN).

b. Press ***---etc--- activity*** (RETURN).

c. Observe either high (H) or low (L) activity on the POD 1 signals.

d. Turn off the Activity Test by repeating step b.

*Observe that you are in the format specification menu.*

## 3. Define a Measurement Mode.

a. Press ***mode wide*** (RETURN).

b. Observe **WIDE_SAMPLE MODE** displayed in the upper left corner of the screen.

*You can change the mode while in the format or trace specification menus.*

## 4. Define Labels.

a. Press ***define***.

b. Type in **CLK**.

c. Press ***pod_1_bit***.

d. Type in **0**.

e. Press (RETURN).

f. Observe that **CLK** has been added to the list of labels.

## 4. Define Labels (Cont'd)

g. Press the following keys and type in the appropriate text to define the remaining signals:

***define*** **LRD** ***pod_1_bit*** **1** (RETURN)

***define*** **LWR** ***pod_1_bit*** **2** (RETURN)

***define*** **ALE** ***pod_1_bit*** **3** (RETURN)

***define*** **S0** ***pod_1_bit*** **4** (RETURN)

***define*** **S1** ***pod_1_bit*** **5** (RETURN)

***define*** **INTR** ***pod_1_bit*** **6** (RETURN)

***define*** **INTA** ***pod_1_bit*** **7** (RETURN)

*If you fill the screen with defined labels, you can roll the screen to observe the other labels by using the* (ROLL UP) *and* (ROLL DOWN) *keyboard keys.*

## 5. Set up trigger specifications.

a. Press ***show tracespec*** (RETURN).

b. Press ***trigger on entering ---etc--- CLK*** **= 1** (RETURN).

*Note that you are in the trace specification menu. The analyzer will trigger when CLK goes high.*

## 6. Execute the measurement.

a. Press ***execute***.

b. Observe that the Timing Diagram is displayed and that the signals are listed from POD1.0 through POD2.7.

c. Press ***display*** and (RETURN) to observe these signals listed by their bit numbers. Press (RETURN) again to return to the original listing. You can also roll the screen by pressing the (ROLL UP), (ROLL DOWN), (NEXT PAGE) and (PREV PAGE) keyboard keys.

*If the relationship between the signals is difficult to see, press* ***magnify x100*** (RETURN).

## 7. Display the defined labels.

a. Press *display ---etc--- CLK* (RETURN).

*Observe that CLK is the only signal label displayed.*

b. Press the following keys in sequence to display all the defined labels.

*display ---etc--- CLK then*
*LRD then ---etc--- LWR then*
*---etc--- ---etc--- ALE then*
*---etc--- ---etc--- S0 then*
*---etc--- ---etc--- S1 then*
*---etc--- ---etc--- INTR then*
*---etc--- ---etc--- INTA* (RETURN)

c. Observe that the defined signals and waveforms are displayed on the timing diagram.

## 8. Observe the Trace List Specification Menu.

a. Press *show tracelist* (RETURN).

b. Observe the display. The POD1 bits are displayed in binary.

c. Display the defined signal labels by repeating part b of step 7.

d. Observe the displayed signals.

## CHOOSING A MEASUREMENT MODE

In Timing Analysis, four measurement modes are available: Wide Sample Mode, Glitch Capture Mode, Dual Threshold Mode and Fast Sample Mode. Use External Clock Mode for State Analysis.

### Wide Sample Mode

Press the following keys:

*mode* *wide* RETURN

This mode is the most versatile and the most commonly used for timing analysis. This is the only mode that allows use of all eight probe input channels. Use this mode for all timing measurements that do not require the specialized functions of the other modes.

### Glitch Capture Mode

Press the following keys:

*mode* *glitch* RETURN

This mode is used for detecting and displaying the occurrence of multiple data transitions between data samples. If more than one transition is detected between samples, this information is stored in a glitch memory and displayed on the screen. A glitch is displayed using a special symbol: a broken vertical bar. In X100, X40 and X20 magnification, a glitch is shown by adjacent broken vertical bars. In X1, X2, X4 and X10 magnifications, a single broken vertical bar is used to indicate a glitch. **This bar also indicates the occurrence of multiple data transitions too close together to be displayed at the selected horizontal magnification.**

### Dual Threshold Mode

Press the following keys:

*mode* *dual* RETURN

This mode is provided for setting up a middle voltage range. The range identifies times when the voltage is between separately-presetable, upper and lower thresholds, as established in the format specification. This mode can be used to identify times when the voltage is between the threshold limits of the TTL $V_{IH\ min}$ and the $V_{IL\ max}$ (as would be caused by excessive fanout or bus contention, for example). This described condition is indicated by a middle level in the trace.

## Fast Sample Mode

Press the following keys:

*mode* *fast* RETURN

This mode is used to obtain very high time resolution of sampled data. The sample rate in this mode is fixed at 400 MHz (2.5 nS period), giving the trace at least twice the resolution of the other modes. This mode also has twice the memory depth (8K) of the other modes.

## External Clock Mode

Press the following keys:

*mode* *ext_clock* RETURN

This mode is used for High Speed State analysis. The external clock probe connects an external clock source to the analyzer. The maximum clock rate acceptable in this mode is 125 MHz (8.0 nS period).

## USING TIMING ANALYSIS

Once you have accessed the analyzer (see page 3-2), you can use timing analysis to monitor and analyze the hardware activity of the system under test.

Four measurement modes are available in the timing analyzer: Wide Sample, Fast Sample, Glitch Capture and Dual Threshold. If you want to choose a mode other than the default mode (Wide Sample), follow these steps:

1. Press *mode*
2. Press any one of the following keys

   _____ *wide glitch dual fast* _____ _____ _____

3. Press RETURN

Now you are ready to work with the Timing Analyzer.

## USING STATE ANALYSIS

Once you have accessed the analyzer (see page 3-2), you can use State Analysis to monitor and analyze software activity of the system under test. You must have the clock probe connected to the external clock source and you must have the data delay module(s) installed. Follow the steps outlined below:

1. Press *mode*
2. Observe the displayed softkeys

   _____ *wide glitch dual fast ext_clock* _____ _____

3. Press *ext_clock*
4. Press RETURN

Now you are ready to work with the High Speed State Analyzer.

**NOTE**

With the High Speed Timing/State Analyzer, you can use state analysis in mostly the same manner as timing analysis, but be aware that some of the commands and capabilities between the two are different.

## DEFINING SIGNAL LABELS

After you have determined the signals to analyze and have connected the probe leads to these signals, enter the Format Specification menu by following these steps:

1. Press *show*
2. Press *format*
3. Press RETURN

Observe the format specification menu. Note that POD 1 is defined and all 8 bits are indicated by an asterisk (*). Up to four pods can be displayed indicating a 32-channel system.

To define labels for the signals connected to the system under test, follow the steps outlined below.

1. Press *define*
2. Type in a name for the label (example: **SYNC**)
3. Press *pod_1_bit*
4. Type in the bit number of POD 1 that you want to define (0-7)
5. Press RETURN

**NOTE**

Observe the listing of signals in the format specification menu. The signal you have just defined will be added to this list.

### Displaying The Defined Label

When you execute a measurement by pressing *execute* the analyzer displays the timing or state diagram. At this point, your defined label will not appear on the screen. To display the label you defined, follow these steps:

1. Press *display*
2. Press *etc*
3. Press the softkey with the label you defined (ex: **SYNC**)
4. Press RETURN

The display will now show the label for the signal you defined. In the format specification, you can rename or delete labels.

**NOTE**

In the Trace List Menu, use the procedure listed above to display the defined label on the screen.

## USING THE SPECIFICATION MENUS

Five specification menus are provided for working with the High Speed Timing/State Analyzer. These menus are: **Trace Specification, Format Specification, Timing or State Diagram, Trace List, and Post_process Specification.**

### Trace Specification

When you press *show tracespec* (RETURN), the following softkeys appear:

*trigger sample _____ mode _____ show execute ---etc---*

When you enter *time_st* (or *timing* when using timing analysis only), the analyzer defaults to this specification menu. Use the Trace Specification to:

1. Set up trigger specifications:
   a. What to trigger on (ex: *any_thing* or *any_glitch* )
   b. Position the trigger at the start or end of trace, or some percentage in between.
2. Set the sample period or rate:
   a. Sample rate is 200 MHz to 2 Hz (period of 5 nS to 500 mS)
      **Fast Sample Mode** - rate is 400 MHz (period of 2.5 nS)
3. Assert trigger output to the development station BNC port

From the trace specification you can also: change the measurement mode, show another menu, execute a measurement, set default specifications, load in and save a configuration, copy the results to a file or printer, and end timing/state.

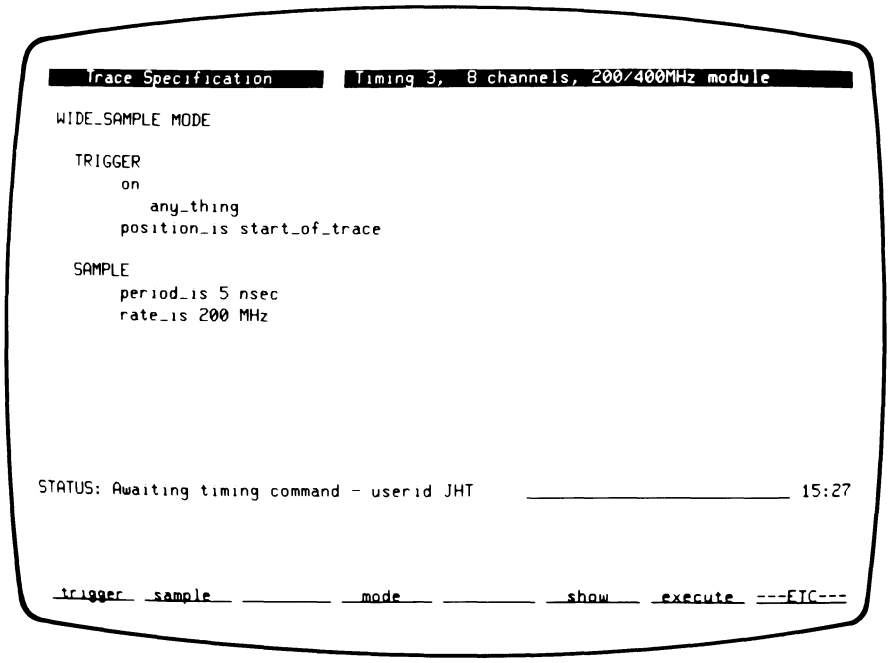



**Figure 3-1. Trace Specification Menu**

## Format Specification

When you press *show format* (RETURN), the following softkeys appear:

*define modify threshold mode clock show execute ---etc---*

Use the Format Specification to:

1. Define labels and set signal polarity
   a. Define individual labels
   b. Define labels to identify multiple signals
2. Set the probe threshold level:
   a. For **Wide Sample, Glitch Capture, Fast** and **External Clock** modes:
      TTL = 1.4 V
      ECL = -1.3 V
      OTHER = -10 V to 10 V (in 0.1 V increments)
   b. For **Dual Threshold** mode:
      TTL = 0.8 V to 2.0 V
      ECL = -1.5 V to -1.1 V
      OTHER = -10 V to 10 V (in 0.1 V increments)
3. Define the clock edge and clock qualifier level in External Clock Mode.

From the format specification, you can also: modify the pod specifications, change the operating mode, show another specification menu, execute a measurement, delete labels, rename labels and pods, set default specifications, run the activity test, load in and save a configuration, copy the results to a file or printer, and end timing/state.

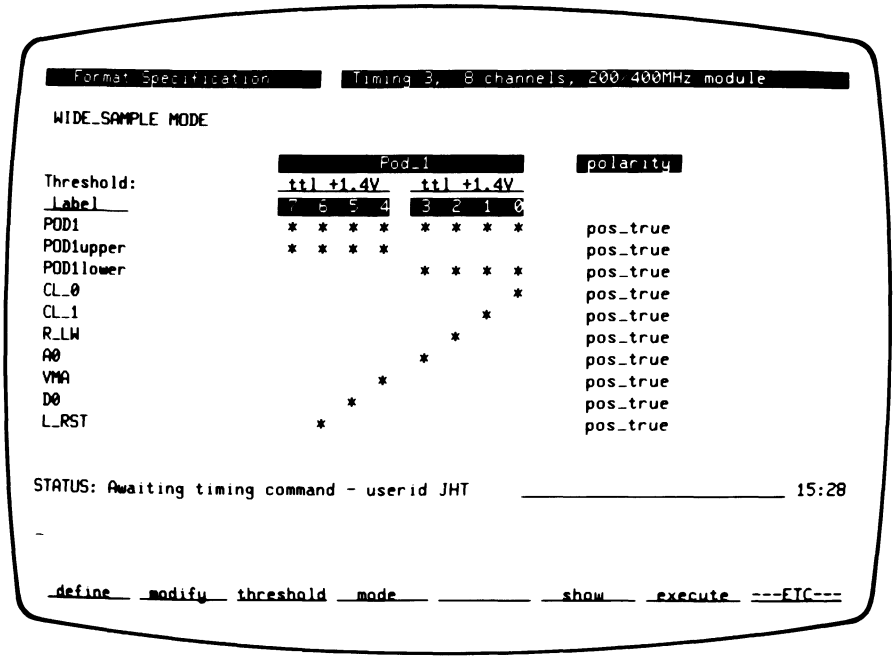



**Figure 3-2. Format Specification Menu**

## Timing And State Diagrams

When you press ***show diagram*** (RETURN), the following softkeys appear:

***display cursor magnify find mark show execute ---etc---***

Use the Timing/State Diagram to:

1. Display the desired signals to be traced (up to 32 signals):
    a. Display a defined label
    b. Display a compare file label
    c. Display the default signals
2. Magnify the screen:
    a. x1, x2, x4, x10, x20, x40 and x100 magnification
    b. Turn the magnify indicator on or off
3. Move the cursor in the diagram:
    a. Use the arrow keys to move the cursor
    b. Change the cursor position with respect to the trigger
       (indicated by the number under "**cursor**" on right side of screen)
    c. Change the display window in trace memory
       (indicated by the upper carat in the "**Trace Memory** = " line)
4. Mark x, o, a, b, c, or d on events in trace memory
5 Find trigger, marks and events in trace memory
6. Indicate time/states or number of states between two points
7. Configure the diagram to display 8 or 16 channels
8. Indicate the data levels of the cursor

The timing and state diagrams are very similar. From the timing and state diagrams you can also show another specification menu, execute a measurement, roll the signals using the arrow keys, change the sample rate (timing diagram only), load in and save a configuration, copy the results to a file or printer, and end timing/state.

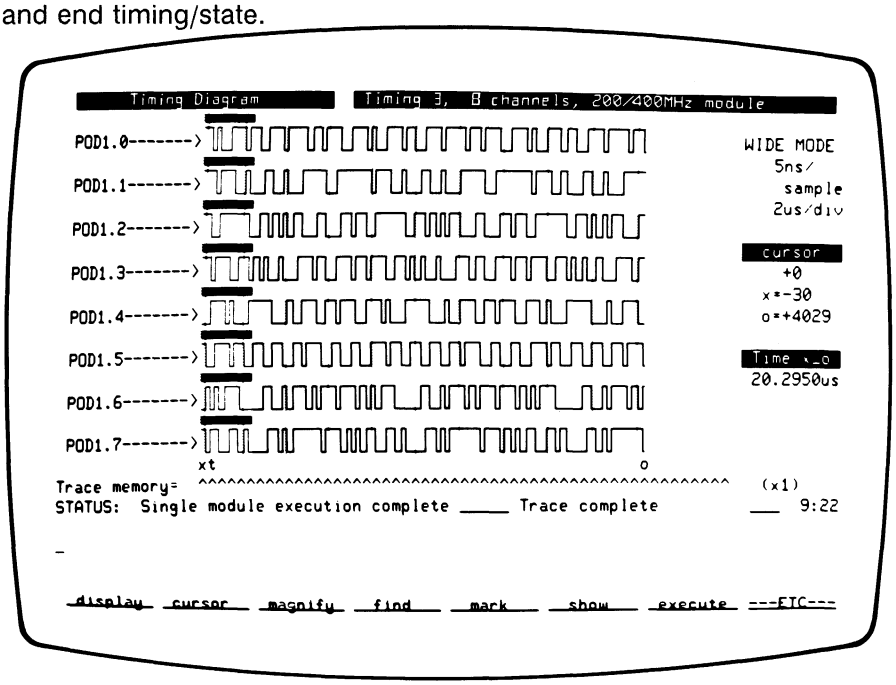



**Figure 3-3. Timing Diagram**

## Trace List

When you press ***show tracelist*** (RETURN), the following softkeys appear:

***display process _____ find mark show execute ---etc---***

Use the Trace List to:

1. Display desired signals:
   a. Display a defined label
   b. Display time/state count
   c. Display mark names
   d. Display the default bits
2. Process for data:
   a. Marked
   b. Samples of a pattern
   c. Greater than some time duration
   d. Sampled some samples after transition of pod bit
   e. Compare faults
3. Find trigger, marks and events in trace memory
4. Mark x, o, a, b, c, or d on events in trace memory
5. Indicate time/states between marks or number of marks between two points

From the trace list you can also show another specification menu, execute a measurement, change the sample period or rate, roll the signals using the arrow keys, load in and save a configuration, copy the results to a file or printer, and end timing/state.

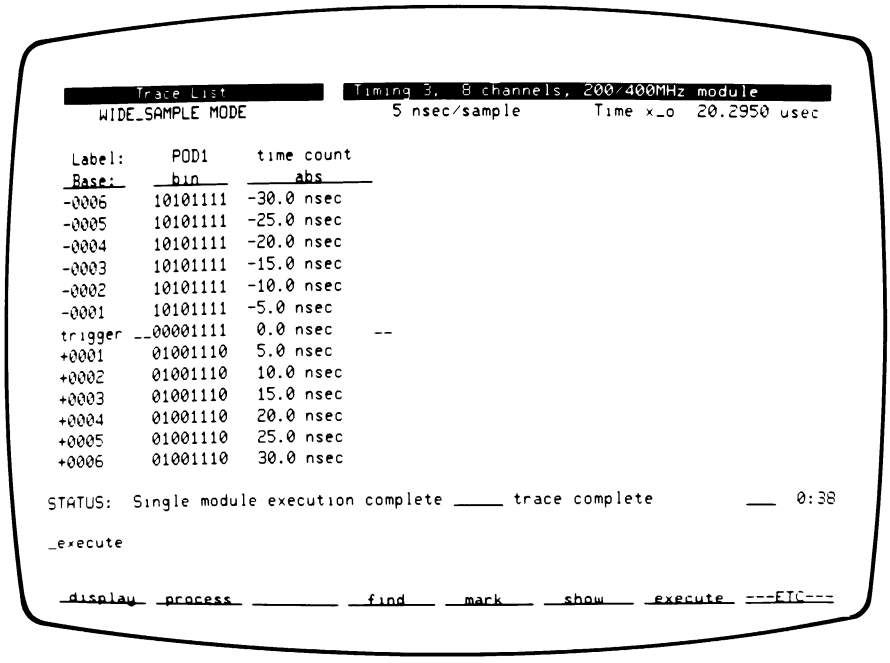



**Figure 3-4. Trace List Menu**

## Post_process Specification

When you press *show postspec* (RETURN), the following softkeys appear:

*halt_rept process compare _____ mark show execute ---etc---*

Use the Post_process Specification to:

1. Halt repetitive execution:
    a. Determined by time between marks
    b. Determined by number of marks
    c. Determined by sequence between marks
    d. Determined by number of runs
    e. Determined by compare
2. Process for data:
    a. Marked
    b. Samples of a pattern
    c. Greater than a defined time duration
    d. Sampled some samples after transition of pod bit
    e. Compare faults
3. Compare a file or current data
4. Mark x, o, a, b, c, and d
5. Mark Statistics:
    a. Always
    b. Determined by time between marks
    c. Determined by number of marks

From the Post_process Specification Menu, you can also: show another specification menu, execute a measurement, set the default configuration, load in and save a configuration, copy the results to a file or printer, and end timing/state.

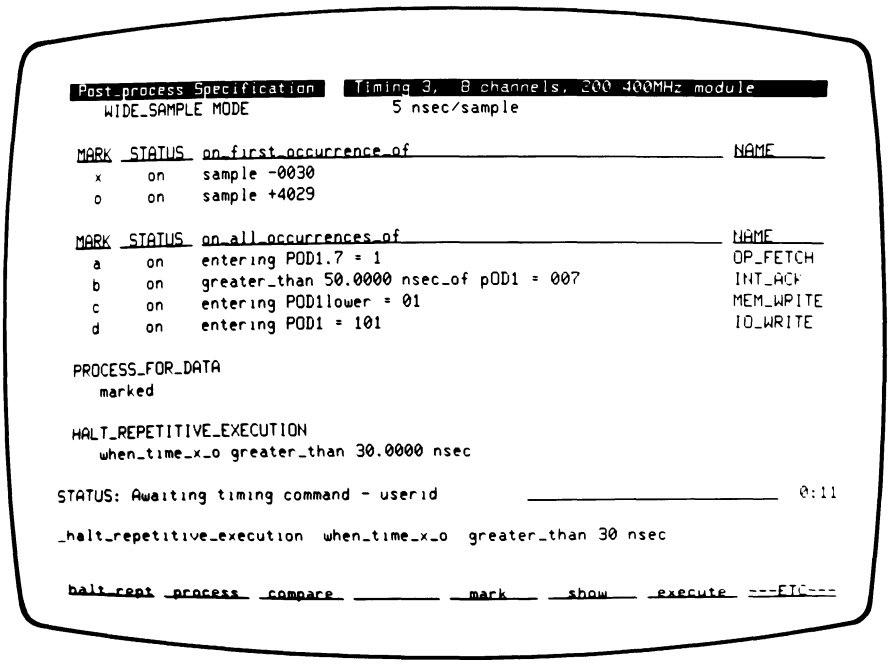



**Figure 3-5. Post_process Specification Menu**

## ENTERING NUMERIC VALUES

You can enter numeric values in four standard bases: binary, decimal, octal and hexadecimal. You must include the base letter after the number you enter, as shown below:

Binary = 1001B
Decimal = 1001D or 1001
Octal = 1001O or 1001Q
Hexadecimal = 1001H

**NOTE**

**If you do not supply a base letter, decimal is assumed.**

You can also enter X as part of your pattern, indicating a "don't care" state. In **Dual Threshold** mode, you can enter M, indicating a middle state. You must precede a number with a zero when that number begins with an X or M. You must precede a hexadecimal number with a zero when that number begins with an A, B, C, D, E or F.

| Base | Correct | | Not Correct |
|---|---|---|---|
| Binary | 01XB | 0M11B | 0MB | MB |
| Decimal | 010D | 0011D | 00D | M or X |
| Octal | 07XO | 0M77O | 07MO | X77O |
| Hexadecimal | 0FXH | 0MFFH | 0MH | MFFH |

**NOTE**

**In decimal you cannot enter M or X.**

# Chapter 4

## CONFIGURING THE ANALYZER

### INTRODUCTION

In this chapter, you will:

* Configure the Analyzer

  Using a Command File

  Using a Configuration File

  Using the Keyboard

## CONFIGURING THE ANALYZER USING A COMMAND FILE

Command files activate the analyzer and perform automatic measurements. You can write your command file into a larger command file that controls activities in several instruments. To write a command file for the High Speed Timing/State Analyzer, get into the monitor (original) level of softkeys and enter the editor mode as shown below.

***edit into*** **CMDFILE (use any command file name)**

Your command file should be configured with the sequence of command lines that you would use to create the desired setup from the monitor level.

**NOTE**

Be sure to use the exact terms that appear on the command line, not the names that appear on the softkeys (ex: use ***mode_is***, not ***mode***).

An example command file is shown below:

```
PARMS & SLOTNUMBER

measurement_system

time_st_SLOTNUMBER

show format_specification

define ALE pod_1_bit 0

show trace_specification

trigger on entering ALE = 1

show timing_diagram

display ALE then POD1

execute
```

When you have completed your command file, press ***end*** and (RETURN).

To run the command file from the system monitor level, type in the command file name (CMDFILE in this example) and press (RETURN).

When you run the command file, the display will prompt you for the slot number of the analyzer control board as shown below:

```
Define parameter &SLOTNUMBER:
```

Type in the card slot number where the analyzer control board is located. The analyzer will then execute the remainder of the command file.

**NOTE**

If the analyzer control board is in slot number 3, you can run the example command file without operator action by replacing the first three lines with the command lines shown below:

**measurement_system**
**time_st_3**

If the analyzer is the only measurement instrument present in the development station, replace the first three lines in the example command file with the single command line shown below:

**time_st**

**NOTE**

**Logging commands to a file is an easy way to create a command file (see page 4-6).**

## Wait Utility

A wait utility exists for use in command files. To use this utility, type in **wait**. Three softkey options will appear: ***meas_comp***, ***<TIMER>*** and ***<RETURN>***.

**meas_comp** - use this parameter after each ***execute*** command to insure that the measurement is completed in the analyzer before the next command in the command file is executed. In your command file, type in: **wait measurement_complete**.

*This parameter is very useful when creating a command file that requires repetitive executions of a measurement, such as* ***halt_repetitive_execution when_time_x_o greater_than 5 nsec****. This insures that the measurement will be executed as many times as is necessary to capture the specified data.*

**<TIMER>** - use this parameter in your command file when calling up a series of displays. This command will cause the selected display to remain on the screen for a determined period of time (in seconds). In your command file, type in: **wait 50**.

**<RETURN>** - use this parameter to suspend execution of the command file. Press any keyboard key to continue execution of the command file. In your command file, type in: **wait**.

**NOTE**

**Press any keyboard key during the wait period of these three parameters to disable the wait function.**

# CONFIGURING THE ANALYZER USING A CONFIGURATION FILE

A configuration file is:

. . .a measurement setup you used to perform previous tests
. . .a measurement setup you stored in a trace file
. . .the default setup

## Getting The Measurement Setup Used To Perform Previous Tests

When you press *end*, the analyzer stores the present measurement specifications. It also continues to run a test according to this specification if the test was in progress when you ended the measurement.

To return the analyzer to the stored setup, enter one of the sets of commands shown below.

Press *meas_sys continue* (RETURN) *time_st* (RETURN)
**OR**
Press *time_st continue* (RETURN)

If a measurement was in process when you pressed *end*, and it has not completed, it may still be running when you reenter.

## Getting A Measurement Configuration From A Trace File

The High Speed Timing/State Analyzer can store measurement configurations in trace files so that you can keep a library of test setups on hand for your measurement needs. The procedure for storing and recovering these measurement configurations is shown below:

1. Set up any desired measurement configuration in your analyzer.

2. Press *configure* and *save_in* and type in **M1**.

   *The present measurement configuration will be saved in a trace file named M1 under the current userid.*

3. Press *with_data* and (RETURN).

   *All of the data in the trace will be saved when measurement configuration M1 is saved.*

4. You can now change the setup. Your original measurement configuration is saved exactly as you last saw it.

## Getting A Measurement Configuration From A Trace File (Cont'd)

5. Press ***end*** to return to the monitor level of softkeys.

6. Press ***time_st***. Type in **M1**. Press (RETURN).

   *This gains access to the analyzer which automatically searches the disc and loads the configuration stored in M1. It will use the data stored in M1 to build the traces and diagrams.*

   **NOTE**

   Use the above method when you are entering the analyzer and want to load a special measurement configuration at the same time.

7. To reload the configuration in file M1 without ending first, press ***configure load_from*** and type in **M1**. Press (RETURN).

   *This causes the analyzer to purge the present measurement setup and load the configuration (with data) you saved in file M1.*

When storing a measurement configuration for future use, you can save time and memory space by omitting ***with_data*** in your command. The analyzer will then save only the measurement configuration.

Use the procedures above to save as many measurement configurations as you desire. Use the write_protect option (***protect***) to protect the configuration files from being rewritten. You can recall and use these configurations for later testing. You can also modify the content of any of these configurations once you have reloaded them into the analyzer.

## Using The Default Configuration

When you press ***time_st*** to gain access to the analyzer, the default measurement setup is automatically entered into the analyzer. Press ***execute*** to run the test.

The default measurement specification may not set up the exact measurement that you want, therefore you can modify the specification. Use softkeys in the format and trace specification menus to set up any measurement desired.

Some default specifications for the High Speed Timing/State Analyzer are listed below.

**Measurement Mode: WIDE_SAMPLE**
**Trigger: on any_thing position_is start_of_trace**
**Sample: rate_is 200 MHz (5 nS period)**

To enter a specification menu, press the ***show*** softkey and one of the specification menu softkeys: ***tracespec, format, postspec, diagram,*** or ***tracelist.***

## CONFIGURING THE ANALYZER USING THE KEYBOARD

Configuring the analyzer using the keyboard consists of manually pressing the desired softkeys and typing in text to set up the desired configuration. The types of keys used in this method are described below.

### Text Softkeys

The text softkeys make entering commands easy by building the commands needed to configure the analyzer. Refer to Appendix B for syntax diagrams that illustrate the use of text softkeys.

### Prompt Softkeys

The prompt softkeys are enclosed in angle brackets and require you to type in text (example: **<FILE>**).

### Utility Keyboard Keys

The utility keyboard keys consist of **RECALL, TAB, CAPSLOCK, INSERT,** and **DELETE.** Use **TAB** to step the cursor forward. Use **SHIFT** and **TAB** together to step the cursor backward.

## LOGGING COMMANDS TO A FILE

Logging commands to a file is an easy way to create a command file. In the monitor level of softkeys, press ---*ETC*--- until *log* appears. Press *log*. Press *to*. Enter a filename that you want to use as your command file (example: **LOGFILE**).

Whether you are in the measurement system level of softkeys or the timing/state analysis level of softkeys, all commands that you enter will be kept to LOGFILE. You will need to edit this command file before using it to make automatic measurements.

To turn off the log commands capability, enter the monitor level of softkeys and press *log off*.

# Chapter 5

## EXAMPLES OF ANALYZER USE

### INTRODUCTION

This chapter consists of examples designed to familiarize you with the High Speed Timing/State Analyzer. The list of features for the High Speed Timing/State Analyzer is included in Chapter 1 - General Information.

Most of the examples in this chapter can be performed using the timing and state portions of the analyzer, however, some of the commands and capabilities differ between the two. Procedures unique to State Analysis are indicated under **STATE ANALYSIS**.

The information in this chapter shows you how to:

* Configure the analyzer for Timing or State Analysis *(mode)*
* Define labels *(define, delete, modify, rename)*
* Test for channel activity *(activity)*
* Set the probe threshold *(threshold)*
* Define the external clock and qualifier *(clock)*
* Set up trigger events *(trigger)*
* Set up trigger position and delay *(trigger)*
* Set the timing sample rate *(sample)*
* Assert the BNC port *(assert)*
* Change the number of channels shown *(diagram)*
* Display channels in the timing/state diagram *(display)*
* Change the timing/state diagram magnification *(magnify)*
* Roll the timing/state diagram *(cursor, <ROLL>)*
* Display channels in the trace list *(display)*
* Roll the trace list *(<ROLL>)*
* Locate events in the trace memory *(find)*
* Determine time/state intervals *(mark, indicate)*
* Automatically mark the trace memory *(mark)*
* Determine time/state interval statistics *(indicate, stats)*
* Determine mark occurrence statistics *(indicate, stats)*
* Indicate date and time and levels at cursor *(indicate)*
* Process the trace_list *(process)*
* Store measurements for later analysis and compare *(configure)*
* Compare a stored measurement with a current measurement *(compare)*
* Automatically compare signals *(compare)*
* Use the extended measurement features *(halt_rept)*
* Copy specifications and results to files or a printer *(copy)*

## CONFIGURE THE ANALYZER FOR TIMING OR STATE ANALYSIS

To configure the analyzer for Timing or State Analysis, use the ***mode*** softkey. You can specify the desired mode from the trace and format specification menus. Enter the command lines below to configure the analyzer in the various modes. External Clock Mode is used for State Analysis.

*mode wide* (RETURN)

This configures the analyzer in the Wide Sample Mode, which is the primary timing analysis mode. Wide Sample Mode allows analysis on all channels from sample rates of 200 MHz to 2 Hz.

*mode glitch* (RETURN)

This configures the analyzer in the Glitch Capture Mode. Glitch Capture Mode detects and displays glitches (multiple transitions between sample points) on channels 0, 1, 2 and 3 of each probe. In this mode, any pulse which is between 3 nS and the defined sample period minus 4 nS wide can be marked as a glitch and triggered on. The sample rates available in this mode range from 100 MHz to 2 Hz.

*mode dual* (RETURN)

This configures the analyzer in the Dual Threshold Mode. This mode compares the sampled data against two thresholds (an upper threshold and a lower threshold) to produce three-level timing waveforms and trace lists. The middle level (M) indicates that the sampled data was between the two threshold voltages specified in the format specification. In the trace list, a middle level is indicated by an "M" in the numerical value. Middle levels can also be triggered on by entering M in the appropriate bit positions. Sample rates can be specified between 200 MHz and 2 Hz. Only channels 0, 1, 2 and 3 of each probe (POD) are active in this mode.

*mode fast* (RETURN)

This configures the analyzer in the Fast Sample Mode. This mode enables channels 0, 1, 2 and 3 of each probe (POD) to capture data at the fixed sampling rate of 400 MHz (2.5 nS sample period). This mode also doubles the amount of memory available for storage of measurement data (8k).

**STATE ANALYSIS**

*mode ext_clock* (RETURN)

This configures the analyzer in the External Clock Mode for running State Analysis. External Clock Mode specifies the external clock input as the clock of the analyzer. The clock and its single input qualifier are specified in the format specification and allow positive or negative edge clocks to be the clock for data capture. When the data is set up at least 4 nS before the clock edge, and stable until the clock edge (0.5 nS hold), the analyzer will capture data up to a maximum clock rate of 125 MHz across all channels on all probes.

## DEFINE LABELS

Defining labels allows you to create names for the signals you are analyzing. You can define labels from the format specification menu. Enter the command lines below (or ones similar) to use this function.

*show format* RETURN

The format specification menu appears. This is where you define labels and redefine pod bits and labels. You also set the signal polarity here.

*define <LABEL> pod_1_bit* **0** RETURN

This creates a user-defined label for bit 0 of pod 1.

*define <LABEL> pod_1_bit* **0** *polarity negative* RETURN

This creates a user-defined label for bit 0 of pod 1 and sets the logic polarity negative true.

*modify <user-defined label>* RETURN

The command line shows the setup for the user-defined label, including polarity. You can now modify this setup.

*delete <LABEL>* RETURN

Use this command to delete labels and pod bits.

*rename <user-defined label> to <LABEL>* RETURN

Use this command to rename user-defined labels and pod bits to new labels.

## TEST FOR CHANNEL ACTIVITY

After you have connected the probe leads to the signals you want to analyze, run this test to observe the voltage activity on each signal. In the format specification, enter the command lines shown below to use this function.

*show format* RETURN

The format specification menu appears. This is where you test for activity on all channels.

*activity* RETURN

A high or low status is indicated under each pod bit number on the display and the word activity is included in the list of labels, indicating the Activity Test is turned on. Press *activity* again to turn off the Activity Test.

## SET THE PROBE THRESHOLD

The user-defined probe threshold levels are provided so that your probes can interpret TTL or ECL levels from the system under test. The threshold can also be set between -10 and +10 volts (in 0.1 volt increments). In Wide Sample, Glitch Capture, Fast Sample and External Clock Modes:

**TTL = 1.4 volts**
**ECL = -1.3 volts**

**In Dual Threshold Mode:**

**TTL = 2.0 volts and 0.8 volts (upper and lower thresholds)**
**ECL = -1.1 volts and -1.5 volts (upper and lower thresholds)**

Enter the following command lines to use the threshold function.

*show format* (RETURN)

The format specification menu appears. This is where you set the probe threshold.

*threshold all_pods all_bits ttl* (RETURN)

This sets all bits in all pods to TTL voltage levels.

*threshold pod_1 all_bits ecl* (RETURN)

This sets all bits of POD 1 to ECL voltage levels.

*threshold pod_1 bits_0_thru_3* **8.6** *volts* (RETURN)

This sets bits 0 through 3 of POD1 to a voltage level of 8.6 volts.

**STATE ANALYSIS**

*threshold clock_pod ecl* (RETURN)

This sets the threshold of the clock pod to ECL voltage levels.

## DEFINE THE EXTERNAL CLOCK AND QUALIFIER

Define the external clock and qualifier from the format specification menu when you use State Analysis. You can set the external clock edge to either rising or falling and set the qualifier to either a high or low level. Enter the command lines below to set up the external clock and qualifier.

*show format* RETURN

The format specification menu appears. This is where you define the external clock and qualifier.

*clock rising* RETURN

This sets the external clock to rising edge.

*clock rising and low_qual* RETURN

This sets up the clock as rising edge and defines the clock qualifier as low level qualify.

## SET UP TRIGGER EVENTS

Set up trigger specifications from the trace specification menu. The extensive trigger capabilities allow you to set your trigger on: transitions of signals (entering, leaving), durations of signals (greater, less), and glitches (Glitch Mode Only). You can also trigger on a sequence of events. In **State Analysis** you can trigger on states or a sequence of states. Enter the command lines shown below to use the trigger functions.

*show tracespec* (RETURN)

The trace specification menu appears. This is where you set up all trigger specifications.

*trigger on entering POD1* = **10H** (RETURN)

The trigger will occur when the bits in POD1 transition to 10H. The value 10H is a user-defined value. Enter any number you desire.

*trigger on greater_than* **50** *nsec_of POD1* = **1** (RETURN)

The trigger will occur after POD1 = 1 for 50 nS or more.

*trigger modify* (RETURN)

The command line shows the present trigger setup. You can now modify the trigger setup.

### GLITCH MODE ONLY

*trigger on any_glitch* (RETURN)

The analyzer will trigger on any glitch.

*trigger on any_glitch on POD1* . **0** *with POD1* = **10H** (RETURN)

The analyzer will trigger on a glitch that occurred on POD1 bit 0 when the value of POD1 is 10H.

### DUAL THRESHOLD MODE ONLY

*trigger on greater_than* **10** *nsec_of POD1* . **0 = 0MB** (RETURN)

The analyzer will trigger after the value of POD1 bit 0 is OMB for 10 nS or more (see Chapter 3 for information on entering numeric values).

## SET UP TRIGGER EVENTS (Cont'd)

### USING TWO OR MORE PODS

***trigger on entering POD1*** **= 10H** ***followed entering POD2*** **= 00H** RETURN

The trigger will occur when two conditions are met: First, POD1 must transition to 10H, then POD2 must transition to 00H. The trigger is marked at the POD2 transition point.

***trigger on entering POD1*** **= 10H** ***when*** **100** ***nsec_of POD2*** **= 00H** RETURN

The trigger will occur when two conditions are met: First, POD2 = 00H for more than 100 nS, then while POD2 = 00H, POD1 must transition to 10H.

***trigger on entering POD1*** **= 10H** ***or POD2*** **= 00H** RETURN

The analyzer will trigger either when POD1 transitions to 10H or POD2 transitions to 00H, providing the non-transitioning pod is in its false state (i.e. POD1 is transitioning to 10H and POD2 < > 00H).

### STATE ANALYSIS

***trigger on state POD1*** **= 10H** RETURN

The trigger will occur when the state or condition of POD1 = 10H.

## SET UP TRIGGER POSITION AND DELAY

You can set the position of the trigger at the start, center or end of the trace. You can also set the trigger to occur some percentage after the start of the trace or some percentage before the end of the trace. You can specify that the trigger be delayed a defined amount of time or a defined number of samples. In External Clock Mode, you can also delay the trigger a defined number of states. Enter the command lines shown below to use the trigger position and delay functions.

*show tracespec* (RETURN)

The trace specification appears. This is where you set up all trigger specifications.

*trigger position center* (RETURN)

When you execute the measurement, the trigger will be placed at the center of the trace.

*trigger position <PERCENT> after_start* (RETURN)

The trigger position is set to a defined percent after the start of the trace.

*trigger <DELAY> ns_after entering POD1* = **10H** (RETURN)

The trigger is delayed a defined amount of time after POD1 transitions to 10H.

*trigger <DELAY> samples_after entering POD1* = **10H** (RETURN)

The trigger will be delayed a defined number of samples after POD1 transitions to 10H.

*trigger <DELAY> delay_clocks_after entering POD1* = **10H** (RETURN)

The analyzer will trigger a defined number of delay clocks after POD1 transitions to 10H. This is used only with an IMB module that can drive the IMB delay clock line.

### STATE ANALYSIS

*trigger <DELAY> states_after state POD1* = **10H** (RETURN)

Use this command in the External Clock Mode to delay the trigger a defined number of states after the state of POD1 = 10H.

## SET THE TIMING SAMPLE RATE

Set the timing sample rate from the trace specification menu. Your system may run at 4 MHz, therefore you can set the timing analyzer sample rate to execute measurements at the rate of your system. Note that both the *rate* and *period* commands are available for your convenience (setting one will automatically change the other).

Enter the command lines shown below to set the timing sample rate.

*show tracespec* (RETURN)

The trace specification menu appears on the display. This is where you set the timing sample rate.

*sample rate* **<FREQ>** *KHz* (RETURN)

The timing sample rate is set to a user-defined frequency between 200 MHz and 2 Hz.

*sample period* **<PERIOD>** *usec* (RETURN)

The timing sample period is set to a user-defined time period between 5 nsec and 500 msec.

## ASSERT THE BNC PORT

To monitor the analyzer trigger event, direct the trigger output to a BNC port at the rear of the development station. Enter the following command lines in the trace specification menu to assert (direct) the trigger output.

*show tracespec* RETURN

The trace specification menu is where you assert the trigger output (BNC port).

*assert bnc_4 on_trigger* RETURN

This directs the trigger output to the development station rear panel BNC connector #4. The BNC port will be asserted on an internal nondelayed trigger (each occurrence of the trigger event) while the analyzer is running.

*assert bnc_4 off* RETURN

This turns off the trigger output to the development station rear panel BNC connector.

## CHANGE THE NUMBER OF CHANNELS SHOWN

In the timing and state diagrams, you can configure the display to show either 8 or 16 channels at one time. Displaying 8 channels at one time allows you to observe the signals in greater detail. By displaying 16 channels at one time you can compare waveforms on a greater number of signals, as with a 16-bit address bus. If your system is 24 or 32 channels, press the NEXT PAGE and PREV PAGE keyboard keys to observe the next set of channels and return to the original set of channels. The ROLL UP and ROLL DOWN keys will roll the timing/state diagram vertically one channel at a time.

Enter the command lines below to set up the desired number of channels.

*show diagram* RETURN

The timing or state diagram is displayed. This is where you set the desired number of channels to be displayed.

*diagram 8_chanls* RETURN

This configures the timing or state diagram to display 8 channels at a time. In this manner, you can observe the displayed signals in greater detail.

*diagram 16_chanls* RETURN

The analyzer timing or state diagram will display 16 channels. Here you can observe activity on a greater number of channels at one time.

## DISPLAY CHANNELS IN THE TIMING/STATE DIAGRAM

After you have defined signal names (labels) for the signals you are analyzing, you can display these labels in the timing/state diagram to have them appear on the screen. You can display up to 32 signals. Enter the following command lines to use the display function.

*show diagram* (RETURN)

The timing or state diagram appears on the screen. Displaying channels in both the timing and state diagrams is accomplished in the same manner. At this point, the signals labeled in the diagram include POD1.0 through PODX.7 (where PODX is the last POD in your system).

*display POD1* (RETURN)

The 8 bits of POD 1 will be displayed at the top of the timing diagram.

*display POD1* **. 0** *then blank then POD1* **. 1** (RETURN)

This will cause POD1 bit 0 to be displayed, followed by a blank line, then followed by POD1 bit 1.

*display* (RETURN)

This will allow you to toggle the diagram labels using the (RETURN) keyboard key. Press (RETURN) to observe default labels and again to observe the user-defined labels.

*display default_bits_in pod_1* (RETURN)

The timing/state diagram shows a list of the default labels/bits in POD1.

*display compare_file <YOUR LABEL>* (RETURN)

The timing/state diagram will display a label you defined in a compare file. This is available only when a compare file has been defined in the post_process specification menu. The *compare_file* softkey can be replaced with the symbol " ~ " to insure that all labels can be entered in the command line.

*display modify* (RETURN)

The command line shows the display setup. You can now modify the display setup.

## CHANGE THE TIMING/STATE DIAGRAM MAGNIFICATION

In the timing or state diagram, use the ***magnify*** softkey to change the time/division specification. This allows you to observe signals in greater detail. You can magnify in seven levels: x1, x2, x4, x10, x20, x40 and x100. Enter the command lines shown below to use the magnify function.

*show diagram* (RETURN)

The timing and state diagrams are where you set the magnification level.

*magnify x10* (RETURN)

This causes the timing or state diagram to expand the area about the cursor by a factor of 10. The time/division or state/division changes correspondingly to always indicate the time or states per division (200 nS/division at sampling period of 5 nS).

*magnify indicator on* (RETURN)

This turns on the magnify indicator (horizontal bar). The indicator shows the area of the display that will appear during the next x10 level of magnification. In x10 magnification, the magnify indicator shows the area of display that will appear in the x100 magnification display. Press ***magnify x100*** to observe this area of magnification.

## ROLL THE TIMING/STATE DIAGRAM

In the timing and state diagrams, use the ***cursor*** and ***<ROLL>*** softkeys to position the cursor in the trace memory. Observe the cursor, which defaults to the trigger location indicated by "t" in the "Trace memory =" line. The location of the cursor is also indicated under the word "cursor" on the right hand side of the diagram.

Enter the command lines below to use the ***cursor*** and ***<ROLL>*** functions.

***show diagram*** (RETURN)

The timing or state diagram is displayed.

***cursor***

The cursor softkey will appear in inverse video, indicating that the cursor is on the waveform. You can now move the cursor by pressing the (→) and (←) keyboard keys (note the change in the cursor position indicated by the number under the word "cursor" on the right side of the diagram). Press ***cursor*** again to turn off the cursor function.

***<ROLL>*** **0** (RETURN)

This causes the cursor to be placed at the trigger point in the timing and state diagrams.

***<ROLL>*** **3** (RETURN)

This places the cursor at the third state stored after the trigger occurred in the timing and state diagrams.

***<ROLL>*** **-10** (RETURN)

This places the cursor at the tenth state stored before the trigger occured in the timing and state diagrams.

(SHIFT)(→)

This command causes the waveform to shift right. Note the cursor position change as indicated by the number under "cursor" on the right side of the diagram. You can press the ***cursor*** softkey and the (→) keyboard key which will shift the diagram left when the cursor reaches the right side of the diagram.

(SHIFT)(←)

This command causes the waveform to shift left. Note the cursor position change as indicated by the number under "cursor" on the right side of the diagram. You can press the ***cursor*** softkey and the (←) keyboard key which will shift the diagram right when the cursor reaches the left side of the diagram.

## DISPLAY CHANNELS IN THE TRACE LIST

After you have defined signal names (labels) for the signals you are analyzing, you can display these labels in the trace list to have them appear on the screen. Enter the following commands to use the display function in the trace list menu.

*show tracelist* RETURN

The trace list is displayed.

*display POD1* RETURN

The trace list will display the bits in POD1.

*display POD1 in_hex* RETURN

The trace list will display the bits of POD1 in hexadecimal. Observe that only two digits are displayed in the POD1 listing.

*display POD1 then time_count absolute* RETURN

The bits of POD1 are displayed in binary, and the absolute time count is displayed. The absolute time count shows the trigger point at time 0 and each stored state at the respective sample period from time 0.

*display default_bits_in pod_1* RETURN

All eight bits of POD 1 are displayed in binary.

*display POD1 then compare_file POD1* RETURN

The trace list will show the current bits in POD1 and the bits of POD1 in the compare file. This function requires that a compare file be specified in the post_process specification. The *compare_file* softkey can be replaced with the " ~ " symbol to insure that all labels can be entered in the command line.

*display POD1 then mark_names* RETURN

The trace list will show the bits of POD1 in binary and will also show the mark names you assigned in the post_process specification. This function requires that you define the mark names in the post_process specification.

## ROLL THE TRACE LIST

In the trace list, use the *<ROLL>* softkey to position the cursor in the trace memory. The cursor is indicated by two dashed lines on either sides of the list of labels. The point in the trace memory where the cursor is located is shown on the left of the screen. Enter the command lines shown below to use the roll function in the trace list.

*show tracelist* (RETURN)

The trace list is displayed.

*<ROLL>* **0** (RETURN)

This causes the cursor to be placed at the trigger point in the trace list.

*<ROLL>* **3** (RETURN)

This places the cursor at the third state stored after the trigger occurred in the trace list.

*<ROLL>* **-10** (RETURN)

This places the cursor at the tenth state stored before the trigger occured in the trace list.

(NEXT PAGE) and (PREV PAGE)

Pressing the (NEXT PAGE) keyboard key causes the next higher set of stored states to be displayed. Pressing the (PREV PAGE) keyboard key will cause the previous set of stored states to be displayed.

(↑) and (↓)

The (↑) keyboard key causes the cursor to move upward. When the cursor has reached the top of the list, the next lower stored state will be displayed. The (↓) keyboard key causes the cursor to move downward. When the cursor has reached the bottom of the list, the next higher stored state will be displayed.

## LOCATE EVENTS IN THE TRACE MEMORY

In the timing/state diagram and trace list, use the *find* softkey to locate events in the trace memory. By using the *find* command, the analyzer will locate events such as the trigger, marks, transitions of signals, and patterns, and display them on the screen. When the specified event is found, the analyzer will position the event at the center of the display.

Enter the command lines below to locate events in the trace memory.

*show diagram* (RETURN)

The timing/state diagram is displayed. Locate events in the timing/state diagram or in the trace list.

*find trigger* (RETURN)

This command positions the cursor at the trigger point and causes the trigger and area about it to be displayed. The trigger point is indicated by a "t" above the "Trace memory=" line. The number under the word cursor indicates where the trigger (and cursor) are positioned.

*find mark_x* (RETURN)

The analyzer will locate the point in the trace memory where mark x is and will display this point on the diagram. The cursor will be located at the mark x point. Locate mark o in the same manner.

*find mark_a* (RETURN)

The analyzer will search from the cursor position for the next occurrence of mark_a in the trace memory and position the cursor at this point. Mark a will be identified by a single broken vertical bar on the diagram. Locate marks b, c and d in the same manner.

*find entering POD1* **. 0 = 0** *thru end* (RETURN)

The analyzer will search through the end of the trace memory for the point where POD1 bit 0 transitions to 0. When the event is found, the analyzer places the cursor at the transition point.

*find greater_than* **20** *nsec_of POD1* **. 0 = 1** (RETURN)

The analyzer will place the cursor at the next occurrence where POD1 bit 0 equals 1 for 20 nano seconds.

*find any_transition on POD1 thru* **30** (RETURN)

The analyzer searches for a transition on POD1. The analyzer will search from the current cursor position through sample number 30.

## DETERMINE TIME/STATE INTERVALS

Determine the time/states between two or more events by using the ***mark x, mark o*** and ***indicate*** keys. Measuring the time/state intervals between specified events is a common use for the High Speed Timing/State Analyzer. This task requires marking the endpoints of the interval to be measured, then reading the time/states between marks shown on the screen. Enter the command lines below to use this function.

***show diagram*** RETURN

Determine time intervals in the timing/state diagram. Time/state intervals can also be determined in the trace list.

***cursor*** → ←

Position the cursor on the first edge of a desired interval (signal) which is to be measured. Note that the ***magnify*** softkey may be helpful to obtain a better view of the desired interval.

***mark x*** RETURN

This causes mark x to be placed at the cursor location.

***cursor*** → ←

Position the cursor on the second edge of a desired interval.

***mark o*** RETURN

This causes mark o to be placed at the cursor location.

***indicate time_interval_x_o*** RETURN

The analyzer will indicate the time between mark x and mark o. This is available in all modes except External Clock Mode.

***indicate states_x_o*** RETURN

The analyzer will indicate the states between mark x and mark o. This command is only available in External Clock Mode.

***mark x on_cursor*** RETURN

The analyzer will place mark x at the current cursor location.

## DETERMINE TIME/STATE INTERVALS (Cont'd)

*mark x on_trigger* RETURN

The analyzer will place mark x at the trigger.

*mark x on_sample* **50** *named START* RETURN

The analyzer will place mark x on sample number 50, and will label this mark START. The mark label (START) will appear in the post_process specification under "NAME" and in the trace list under "mark names".

*mark x off* RETURN

This command turns off the mark x indicator. The definition of the mark location is retained.

*mark x off default* RETURN

This command turns off the mark indicator and defaults the definition of the mark location to the start of trace.

*mark x on* RETURN

This turns on the mark indicator at the last specified definition.

## AUTOMATICALLY MARK THE TRACE MEMORY

By using the *mark* command, finding and marking events is accomplished in one step. When automatically marking the trace memory, specified events are searched for in the trace data. When the events are found, they are labeled (x, o, a, b, c, or d) to indicate where they are located in the trace data. Mark x and mark o can each mark one event. Marks a, b, c and d can each mark an occurrence of a defined event, up to a maximum of 511 marked events total. The marking occurs after the analyzer has completed an execution and the execution has filled the trace memory with data. The post_process specification shows the current mark commands which will occur after each execution.

Enter the command lines shown below to automatically mark the trace memory.

***mark x on_first entering POD1 . 6 = 1 after trigger*** (RETURN)

This command causes the analyzer to place an "x" at the first occurrence of POD 1 bit 6 transitioning to 1, after the trigger.

***mark o on_first entering POD1 . 6 = 1 after mark_x*** (RETURN)

This command causes the analyzer to place an "o" at the first occurrence of POD1 bit 6 transitioning to 1, after mark x.

***mark x on_first leaving CLOCK = 1 after cursor named CLOCK*** (RETURN)

Mark x will be placed at the first point after the cursor where CLOCK transitions out of a 1. The names of marks can be displayed in the trace list, and appear in the post_process specification.

***mark o on_first any_trans on DATA before mark_x*** (RETURN)

The analyzer will place mark o on the first occurrence of any transition on a label called DATA, before mark_x.

***mark a on_all greater 5 nsec POD1 . 6 = 1 before 200*** (RETURN)

The analyzer will place an "a" at all occurrences before sample 200 where POD1 bit 6 has transitioned to 1 for 5 nS or more. A vertical row of marks will be placed at the points where the analyzer located the mark_a events. Turn these marks off by entering ***mark a off*** and on again by ***mark a on.*** Remove the mark a definition totally by entering ***mark a off default.*** Marks b, c and d function in the same manner as mark a.

***mark a on_all_occurrences_of compare_faults*** (RETURN)

The analyzer will mark each compare fault (differences between the compare file label and the currently defined label). In the timing/state diagram, each compare fault will be marked above the signal. In the trace list, the marks will be added to the samples where the compare faults occurred.

## DETERMINE TIME/STATE INTERVAL STATISTICS

Time interval measurements can be accumulated for statistical evaluation of time intervals. In External Clock Mode, state interval measurements can be accumulated for statistical evaluation of state intervals. To accomplish this, after each execution a maximum, minimum, mean and standard deviation of the specified interval is computed and displayed. Enter the command lines below to determine the time interval statistics.

***show diagram*** (RETURN)

The timing/state diagram is displayed. Determining time interval statistics can be done in the timing/state diagram or the trace list. In External Clock Mode, state interval statistics can be calculated by ***indicate states_x_o***, just as the time interval statistics calculation described below.

***indicate time_x_o mean_and_standard_deviation*** (RETURN)

This command turns on the statistics analysis capability. By executing the measurement repetitively, the statistics can be calculated. The time interval from mark_x to mark_o will be indicated on the screen. The mean is the average time between mark_x and mark_o after a number of runs. The standard deviation indicates the standard deviation from the mean for the same number of runs.

***indicate time_x_o maximum_and_minimum*** (RETURN)

This command also turns on the statistics analysis capability. The ***maximum*** is the maximum interval found over a number of runs. The ***minimum*** is the minimum interval found over the same number of runs. Two of the four statistics parameters of a number of runs can be displayed at one time. You can observe all four statistics parameters at one time by copying the statistics to a file.

***show postspec*** (RETURN)

The Post_process specification is displayed. Set up the statistics specification in this menu.

***stats when_time_x_o less_than*** **40** ***nsec*** (RETURN)

This command qualifies the statistics to be calculated only if the statistics are in a specified range. This is helpful for making setup and hold time measurements.

***statistics log_file <FILE>*** (RETURN)

This command logs the statistics to the specified file after every execution. One of the statistics operations must be indicated before this command will operate (***max_min*** or ***mean_stdv***).

## DETERMINE MARK OCCURRENCE STATISTICS

To locate and count events, it is helpful to mark each occurrence of a specified event. In the High Speed Timing/State Analyzer, four kinds of events can be marked multiple times, up to a total of 511 marks. Enter the command lines below to use the mark occurrence statistics function.

*show diagram* RETURN

The timing/state diagram is displayed. Mark occurrence statistics are determined in the timing/state diagram or the trace list.

*indicate marks mean_and_standard_deviation* RETURN

The number of marks between mark_x and mark_o will be indicated on the screen. The number of marks represents the number of times all of the marks occurred. The mean is the average number of marks recorded after a number of runs. The standard deviation indicates the standard deviation from the mean for the same number of runs.

*indicate marks maximum_and_minimum* RETURN

This command also turns on the statistics analysis capability. The ***maximum*** is the maximum number of marks found over a number of runs. The ***minimum*** is the minimum number of marks found over the same number of runs. Two of the four statistics parameters of a number of runs can be displayed at one time. You can observe all four statistics parameters at one time by copying the statistics to a file.

*show postspec* RETURN

The Post_process specification menu is displayed. Set up the statistics specification in this menu.

*stats when_marks_x_o less_than* **25** RETURN

This command qualifies the statistics to be calculated only if the statistics are in a specified range.

*statistics log_file* **<FILE>** RETURN

This command logs the statistics to the specified file after every execution. One of the statistics operations must be indicated before this command will operate (***max_min*** or ***mean_stdv***).

## INDICATE DATE AND TIME AND LEVELS AT CURSOR

When storing measurements, you can indicate the current date and time in the timing/state diagram and the trace list. This is an easy way to recall when the measurement was made. In the timing/state diagram, you can also indicate the signal levels at the point where the cursor is located in the trace memory.

***indicate date_and_time on*** RETURN

This command causes the date and time to be displayed just below the list of labels in the diagram and below the last sample in the trace list. This is the date and time of when the measurement was completed. Turn the date and time indicator off by pressing ***indicate date_and_time off***.

***indicate levels_at_cursor on*** RETURN

The analyzer will display "h", "l", "g" or "m" for high, low, glitch (multiple transition) or middle levels, respectively, at the point where the cursor is located in the trace memory. The "h", "l", "g" and "m" are located in the label lines.

## PROCESS THE TRACE LIST

The trace list is an alternative method of observing the measurement results. By processing the trace list, the amount of data displayed is reduced, which makes it easier to read the significant information in the trace list. Enter the command lines shown below to process the trace list.

*show tracelist* (RETURN)

The trace list is displayed. Use the *process* command to process the trace list.

*process_for_data marked* (RETURN)

This command causes the analyzer to display only the marked events in the trace list. All events that do not meet the mark qualifications will be removed from the trace list.

*process_for_data samples_of POD1* **. 6 = 1** (RETURN)

The trace list will show only the occurrences of POD1 bit 6 when that bit equals 1.

*process_for_data states_of POD1* **. 6 = 1** (RETURN)

The trace list will show only the states of POD 1 bit 6 equal to 1.

*process_for_data greater_than* **40** *nsec on POD1* **. 1** (RETURN)

The trace list will show one sample for each occurrence where POD1 bit 1 is in a constant state for 40 nsec or more. All other samples that do not meet these qualifications will not be included in the trace list.

*process_for_data sampled* **2** *samples_before pos_transition_on POD1* **. 1** (RETURN)

The analyzer will store each sample that occurs 2 samples before the positive transition on POD1 bit 1. All other samples that do not meet these specifications will not be included in the trace list.

*process_for_data compare_faults* (RETURN)

The trace list will show only the compare faults (differences between the current pod bits and the compare file pod bits).

*process_for_data modify* (RETURN)

Use this command to make easy modifications to the process_for_data setup.

*process_for_data off* (RETURN)

This command will cause the process_for_data setup to be turned off.

## STORE MEASUREMENTS FOR LATER ANALYSIS AND COMPARE

The High Speed Timing/State Analyzer allows you to store the current configuration and the captured data in a file for later analysis. When the analyzer is reloaded with the stored file, the data can be processed as if it had not left the analyzer. The stored configuration can also be compared with the current configuration. Enter the command lines below in any of the specification menus to store measurements.

***configure save_in*** **TEST1**

The analyzer will save the present analyzer configuration in a file named TEST1.

***configure save_in*** **TEST1** ***with_data*** RETURN

The analyzer will store the configuration with the measurement data in a file named TEST1.

***configure save_in*** **TEST2** ***with_data protect*** RETURN

The present analyzer configuration and the measurement data will be stored in a file named TEST2. This file will also be write protected (you will not be able to save another configuration to this filename until the original file has been purged).

***configure load_from*** **TEST2** RETURN

The analyzer will load the measurement configuration stored in TEST2. The measurement data will also be loaded into the analyzer if it was saved with the configuration in TEST2.

## COMPARE A STORED MEASUREMENT WITH A CURRENT MEASUREMENT

By storing a measurement configuration in a file, you can retrieve and display previously analyzed data with current data. This function requires that you specify a compare file in the Post_process specification menu before trying to compare measurements. The compare file can be any file where you stored a previous measurement configuration with data. **The compare file configuration must be similar to the current analyzer configuration (mode, trigger position and sample period must be the same for both).**

Enter the command lines below to use the compare function.

*show postspec* (RETURN)

The Post_process specification menu is where you define the compare file.

*compare file_is* **TEST2** (RETURN)

This defines a compare file to be used for comparing previously stored measurements with current measurements. The file TEST2 can be any file you stored containing the previously stored measurement configuration with data.

*show diagram* (RETURN)

The timing/state diagram appears. In the timing/state diagram, display the signals to be compared. The signals to be compared can also be displayed in the trace list.

*display POD1* **. 0** *then compare_file POD1* **. 0** (RETURN)

Two signals will be displayed in the timing/state diagram. POD1 bit 0 in the current measurement configuration will be displayed, then POD1 bit 0 in the compare file will be displayed. The compare file label is distinguished from the current label with an "x" included at the end of the label.

*display POD1* **. 0** *then* **~** *POD1* **. 0** (RETURN)

This is identical to the command above, but " ~ " replaces the *compare_file* softkey. This symbol is used to shorten the command line.

## AUTOMATICALLY COMPARE SIGNALS

The High Speed Timing/State Analyzer allows you to automatically compare signals just by entering the signals (pod bits) you want to compare. This function requires that you have already specified a compare file. Enter the following command lines to automatically compare signals.

*show postspec* (RETURN)

The post_process specification menu appears. This is where signals are automatically compared.

*compare all_bits* (RETURN)

All of the bits in the specified pods will be compared. The status line will indicate the number of compare faults found.

*compare POD1 to_compare_file POD1* (RETURN)

The bits of POD1 in the current measurement will be compared to the bits in POD1 of the compare file. Any compare faults will be indicated on the status line. The *compare_file* command can be replaced with " ~ " to shorten the command line.

*compare POD1* . 0 *to POD1* . 0 *and POD1* . 1 *to POD1* . 1 (RETURN)

The analyzer will compare POD1 bit 0 of the current measurement to POD1 bit 0 of the compare file, and will also compare POD1 bit 1 of the current measurement to POD1 bit 1 of the compare file. You must display these signals in the timing/state diagram and trace list to have them appear on the screen.

*compare POD1 to POD1 when POD1* . 0 = 1 (RETURN)

The analyzer will compare POD1 of the current measurement to POD1 of the compare file when bit 0 of POD1 = 1.

*compare POD1* . 4 *to POD1* . 4 *from cursor thru end* (RETURN)

The analyzer will compare POD1 bit 4 of the current measurement to POD1 bit 4 of the compare file. The compare range is from the cursor through the end of trace memory.

*compare consecutive_faults_allowed_is* 2 (RETURN)

The analyzer will compare the current measurement to the compare file according to the last compare specification, and will only indicate faults if more than 2 consecutive faults occur. This is useful for timing analysis compare.

*compare modify* (RETURN)

The compare command line is displayed for easy modification.

## USE THE EXTENDED MEASUREMENT FEATURES

The *halt_repetitive_execution* command is provided so that you can automatically halt a repetitive execution (*execute repetitively*) on a duration, count, sequence or compare qualify. The halt_repetitive_execution setup is made in the Post_process specification. Enter the command lines below to use this function.

*show postspec* (RETURN)

This is the only menu where the halt specification can be made.

*halt_repetitive_execution when_runs_equals* **10** (RETURN)

The analyzer will halt a repetitive execution after the tenth run (execution).

*halt_repetitive_execution when_time_x_o greater_than* **10** *usec* (RETURN)

This command will cause the analyzer to halt the repetitive execution when the time between the x and o marks is greater than 10 usec.

*halt_repetitive_execution when_marks_x_o less_than* **20** (RETURN)

The repetitive execution will be halted when the analyzer identifies less than 20 marked events occurring between the x and o marks.

*halt_repetitive_execution when_sequence_x_o mark_a then mark_b* (RETURN)

The repetitive execution will be halted when the defined sequence (mark_a then mark_b) occurs.

*halt_repetitive_execution when_compare equal* (RETURN)

The analyzer will halt a measurement when the bits being compared are equal.

*halt_repetitive_execution off* (RETURN)

This command turns off the halt function.

*halt_repetitive_execution modify* (RETURN)

This command causes the present halt repetitive execution command line to be displayed. You can now make changes easily to the halt function without having to reenter the entire specification.

## COPY SPECIFICATIONS AND RESULTS TO FILES OR A PRINTER

In the High Speed Timing/State Analyzer, you can copy the trace, format, and post_process specifications and the trace list to a file or printer. You can copy the timing/state diagram to a printer (if you copy the timing/state diagram to a file, the results will not be as you see them in the timing diagram unless you have a screen printer card and a graphics printer).

*copy tracespec to printer* (RETURN)

The trace specification menu will be copied to the printer.

*copy format to* **FILE1** (RETURN)

The format specification menu will be copied to a file named FILE1.

*copy postspec to* **FILE2 <HEADER>** (RETURN)

The post_process specification will be copied to a file named FILE2. The <HEADER> is information that you enter, and will appear on the second line of the specification when copied to a file. Press the *<HEADER>* softkey to observe the delimiters.

*copy tracelist thru end to* **FILE1** *append* (RETURN)

The tracelist will be appended to a file named FILE1. Only the samples from the cursor through the end of the trace list will be copied to FILE1.

*copy diagram to printer* **'HEADER FOR DIAGRAM'** (RETURN)

The timing/state diagram will be copied to the printer with the header information on the second line. The header information reads HEADER FOR DIAGRAM.

*copy diagram graphic to printer* (RETURN)

The timing diagram will be copied to the printer. This command requires that you have a screen printer card and a graphics printer attached to your development station.

In the timing/state diagram and the trace list, you can also copy the statistics and measurement data in binary to a file. This option allows you to store the captured data for later analysis. The statistics is a listing of the last recorded statistics results (time/state/marks between mark x and mark o and associated maximum, minimum, mean and standard deviation). The measurement data is the captured data in a binary format suitable for analysis by an external processor. See Appendix C (Storing Measurement Data In Binary) for the format.

# Chapter 6

## INTERACTING WITH OTHER ANALYZERS

### INTRODUCTION

This chapter contains information on:

* Making Interactive Measurements
* Using Two Timing Analyzers To Make Measurements
* Using A Timing Analyzer And A State Analyzer To Make Measurements
* Using The Restart Function

## WHAT ARE INTERACTIVE MEASUREMENTS?

Interactive measurements are coordinated measurements made between two or more analyzers. These measurements are made through the intermodule bus (IMB) cable connected to the IMB connectors on each analyzer control board. To obtain interactive measurements, enter the analyzer trace specification and enter the desired interactions.

The High Speed Timing/State Analyzer can receive or drive enable levels or triggers between other analyzers in the system. It can also receive a delay clock signal from the other analysis modules on the IMB.

## MAKING INTERACTIVE MEASUREMENTS

All of the selections affecting interactive measurements for the High Speed Timing/State Analyzer are made in the trace specification menu.

The five IMB signals include: **Master Enable, Trigger Enable, Trigger, Storage Enable,** and **Delay Clock.**

The **Master Enable** line is shared by all analysis instruments included in a measurement. Only the *execute/halt* key or another subsystem designated to participate in the measurement can drive this line.

The **Trigger Enable** line can alternate between true and false during a measurement to allow the controlling analyzer to window the program activity in other analyzers where trigger recognition can occur. The High Speed Timing/State Analyzer can drive this line or receive it from some other analyzer on the intermodule bus by the following commands:

*trigger enable received*

*trigger enable driven on_trigger*

*trigger enable driven on <PATTERN>*

*trigger enable driven on <DELAYED PATTERN>*

The **Storage Enable** line determines whether or not the receiving subsystems are allowed to store information. This line can alternate between true and false during a measurement to window the activity to be stored. The High Speed Timing/State Analyzer does not receive or drive this line and therefore has no interaction with the storage enable function.

The **Delay Clock** line allows analyzers such as the High Speed Timing/State Analyzer to delay its trace point by a selected number of clocks. The High Speed Timing/State Analyzer can only receive clocks from this line while other analyzers (such as the 10 MHz State Analyzer) can send them.

## MAKING INTERACTIVE MEASUREMENTS (Cont'd)

The High Speed Timing/State Analyzer can drive the **Trigger** line when it recognizes its trigger specification. It can also receive **trigger**, which will trigger itself, or it can both receive and drive **trigger**. The trigger line transmits to another analyzer the fact that a trigger event has occurred in the analyzer sending the trigger. The trigger line can have more than one designated driver. Some commands are shown below.

***trigger received***

***trigger driven_on <PATTERN> or <DELAYED PATTERN>***

***trigger received_or_driven on <PATTERN> or <DELAYED PATTERN>***

* ***trigger enable restart***

* *This mode requires the High Speed Timing/State Analyzer and an additional state analyzer to operate together on the intermodule bus and* ***trigger enable received*** *and* ***trigger enable restart*** *both to be specified.*

**NOTE**

When the High Speed Timing/State Analyzer's trigger is ***received***, ***driven_on***, or ***received_or_driven***, the "t" which indicates the trigger position on the display is replaced by an "i" ("i" indicates the point in time on the timing diagram when the IMB trigger line switched to true).

## USING TWO TIMING ANALYZERS TO MAKE MEASUREMENTS

The following information tells you how to make interactive measurements using the High Speed Timing Analyzer and an additional timing analyzer.

1. Press ***meas_sys***. Press ***continue*** (if you want the timing analyzers to continue using their former test setups). Press RETURN.

2. Observe the timing analyzer identifiers on the softkey label line (your line will be similar):

***time_st_1 timing_4 _____ _____ _____ _____ _____ end***

3. Press the timing softkey for the analyzer you want to set up first. Add the file name if you want to load a configuration file from memory. Press RETURN.

4. Make any desired modifications to the configuration. Add the desired interaction for this analyzer. Press ***end*** and RETURN.

*The measurement system display will return and will identify the interaction you assigned to this timing analyzer.*

5. Press the other ***timing*** softkey. Determine whether you want to use the default setup or load in a configuration file from memory. Press RETURN.

6. Make any desired changes to the measurement setup for this analyzer. Add the desired interaction to the specification. Press RETURN.

7. Press ***end***. Press RETURN to reenter the measurement system level of display. (This display will show the interactions you specified for both timing analyzers.)

**NOTE**

You can ***execute*** or ***halt*** interactive measurements while the system displays the measurement system monitor or while it displays specifications from any one of the analyzers.

8. Press ***execute***. Press RETURN. (This starts both analyzers running according to their own measurement configurations, provided they both have some interaction specified).

**NOTE**

You can select any of the analyzers in your development station and observe their present specifications. You can not execute any measurements as long as your interactive measurement is in process.

## USING TWO TIMING ANALYZERS TO MAKE MEASUREMENTS (Cont'd)

9. Press ***end***. Press RETURN.

*You can now use the development station for nonanalysis activity while the analyzers execute their measurements. Both analyzers will continue to run measurements according to their specifications and will save data in their memories.*

10. To observe the results of the timing analysis measurements, press ***meas_sys*** then press ***continue***. Press RETURN.

11. Press the timing softkey for the analyzer you want to observe.

*This again gains access to the analyzer you selected without reloading the module. The measurement you assigned to the timing analyzer will still be running if not already completed. You can select displays and use the analyzer's capabilities as though it were the only analyzer in the development station.*

12. When you are finished, press ***end***. Press RETURN.

13. You can check the status of the measurement in the other timing analyzer by pressing the respective timing softkey and the RETURN key.

## USING A STATE AND A TIMING ANALYZER TO MAKE MEASUREMENTS

Using a Timing Analyzer and a State Analyzer together to make interactive measurements requires completing the following steps:

1. Set up the State Analyzer (10 MHz State Analyzer, for example)
2. Set up the High Speed Timing Analyzer
3. Execute the measurement

As an example, assume the system under test executes software normally within certain routines, but occasionally jumps out of these routines and fails. You can set up the Timing and State Analyzers to determine the cause of the failure.

### State Analyzer Setup

Set up the 10 MHz State Analyzer (for example) to trigger a trace when it detects any address outside the range of normal routines and at the same time, trigger a timing measurement in the High Speed Timing Analyzer. Use the following setup or one similar:

*trigger int/drive on ADDRESS <> range FUNCTION* RETURN

*store on any_state* RETURN

*This sets up the State Analyzer to trigger on an address outside the range of the function being performed.*

### Timing Analyzer Setup

Set up the High Speed Timing Analyzer to receive the trigger from the intermodule bus and to position the trigger near the center of the trace memory using the commands shown below:

*trigger received* RETURN

*trigger position center* RETURN

### Execute The Measurement

Press the following keys:

*execute* RETURN

This will begin execution of the measurement. When your software under test exits the range of addresses where the function is being performed, the State Analyzer will trigger a trace and will also trigger a measurement in the High Speed Timing Analyzer.

## Execute The Measurement (Cont'd)

The State Analyzer trace list will show which addresses were being executed when the error condition occurred, and which new addresses were entered after the error condition.

You may need to make a series of measurements to determine the cause of the error. By doing this you can determine which components of the error activity are consistent and which components fail randomly. This will lead you to the cause of the problem.

**NOTE**

If the trigger is driven or received from the IMB, the trigger location on the display is marked with "i" (instead of "t") to indicate the point in memory where the trigger signal occurred on the intermodule bus.

When using "*trigger driven on . . .*", note that the trigger condition occurred 50 nS before the trigger signal was placed on the bus. This will cause the "i" indicator to be placed 50 nS later than the true trigger point on the display.

## USING THE RESTART FUNCTION

Using the trigger enable restart function requires the following:

1. A High Speed Timing Analyzer and a state analyzer (10 MHz State Analyzer, for example) must be operating together on the IMB.

2. The *trigger enable restart* command must be specified.

In this mode, while the Timing Analyzer performs a measurement, it also monitors the trigger enable line. If the State Analyzer causes the trigger enable line to switch to a false state, the Timing Analyzer will discontinue the current measurement. When the trigger enable line is switched back to a true state, the Timing Analyzer begins to perform a new measurement according to its present measurement setup.

This process continues until the State Analyzer sends a trigger signal on the IMB trigger line. The purpose of this command is to cause the analyzer to capture High Speed Timing data near a point in time that may lead to faulty state flow.

# Appendix A

## SOFTKEY GLOSSARY

### INTRODUCTION

This glossary contains keywords which appear on the softkeys and in commands in the High Speed Timing/State Analyzer. The specification menus where each term appears are indicated by the following abbreviations:

**FS** = Format Specification
**PP** = Post_process Specification
**SD** = State Diagram
**TD** = Timing Diagram
**TL** = Trace List
**TS** = Trace Specification

**Command Line Label** **Softkey Label**

***absolute*** ***absolute***

Used with *display* command to define that the time displayed will be total time between the sample on each line and the time when the trigger occurred.

**TL -** ***display POD1 then time_count absolute***

***activity_test*** ***activity***

Used to display voltage activity on the signals being probed.

**FS -** ***activity_test***

***after*** ***after***

Used with the *mark* command to define a range of memory where the identifying mark(s) is to be assigned.

**SD, TD, TL, PP -** ***mark_x on_first_occurrence_of entering POD1 .6 = 1 after trigger***

***all*** ***all***

Used with the *copy trace_list* command sequence to define that the entire list is to be copied to the destination file or device. Used with the *find* command to define the entire memory as the region to be searched.

**TL -** ***copy trace_list all to printer***
**TL -** ***find mark_a all***
**TD, TL -** ***find any_transition all***

***all_bits*** ***all_bits***

Used with *threshold* command to define that this threshold voltage specification should apply to bits 0..7 in the pod being specified. Used with *compare* to define signals to be compared with compare_file signals.

**FS -** ***threshold pod_1 all_bits ecl***
**PP -** ***compare POD1 to_compare_file POD1***

***all_default_bits_in_all_pods*** ***all_bits***

Used with the *compare* command in the Post_process specification to compare all the pod bits of the current measurement to all the pod bits of the compare file. This command requires that a compare file be specified.

**PP -** ***compare all_bits when POD1 . 0 = 1***

***all_labels*** ***all_label***

Used with the *delete* command to erase all label definitions. Only labels not used in any specification will be deleted.

**FS -** ***delete all_labels***

***all_pods*** ***all_pods***

Used with the *threshold* command to define that this threshold voltage specification applies to all pods in the analysis system.

**FS -** ***threshold all_pods all_bits ecl***

***all_specifications*** ***all_specs***

Used with the *copy* command to define that the Trace and Format Specifications and Trace List are to be copied to the destination file or device. Used with the *default* command to restore the Trace, Format, and Post_process Specifications, and the Trace List and Timing Diagram display formats to their default conditions and labels.

**TS, FS, PP, TD, TL -** ***copy all_specifications to printer***

**TS, FS, PP -** ***default all_specifications***

***always*** ***always***

Used with the *trigger enable* sequence to reset any interactive trigger specification so as not to listen to the Low Trigger Enable line in the Inter Module Bus. This will clear any trigger enable out of the specification, causing it to disappear from the display.

**TS -** ***trigger enable always***

**PP -** ***statistics always***

***and*** ***and***

Used with a number of commands to extend definitions.

**TS -** ***trigger on entering POD1.0 = 0 and POD1.1 = 1***

**FS -** ***threshold pod_1 bits_0_thru_3 ecl and pod_1 bits_4_thru_7 ttl***

**FS -** ***define LABEL pod_1_bit 0 and pod_1_bit 1***

***any_glitch*** ***any_glch***

Used with the *trigger* command during Glitch Capture Mode to include a glitch or multiple glitches in the trigger definition. Used with the *find* command during Glitch Capture Mode to find a sample where a glitch or multiple glitches were detected during the trace measurement.

**TS -** ***trigger on any_glitch on POD1lower.0***

**SD, TD, TL -** ***find any_glitch on POD1lower.1 or_on POD1lower.2***

**TD, TL -** ***mark o on_first_occurrence_of any_glitch***

***any_mark*** ***any_mark***

Used with the *halt_repetitive_execution* command to designate a don't-care member of a sequence of occurrences which, when found in order, will cause the analyzer to halt the repetitive executions in progress.

**PP -** ***halt_repetitive_executions when_sequence_x_o mark_a then mark_a then any_mark then_not mark_c***

***any_thing*** ***any_thing***

Used with the *trigger* command to establish a "don't-care" trigger condition which will be satisfied by any probe input pattern.

**TS -** ***trigger on any_thing***

***any_transition*** ***any_trans***

Used with the *mark* command to set up a mark on the first occurrence or all occurrences of any transition.

**PP, SD, TD -** ***mark x on_first_occurrence of any_transition***

***append*** ***append***

Used with the *copy* command to append the analyzer specification being saved to the end of a file.

**FS, PP, SD, TD, TL, TS -** ***copy tracespec to <file> append***

***assert*** ***assert***

Command used to turn on or off the trigger output to the rear panel BNC connector.

**TS -** ***assert bnc_port_4 on_internal_nondelayed_trigger***

***before*** ***before***

Used with the *mark* command to set up a mark to occur before a defined condition.

**PP, SD, TD, TL -** ***mark a on_all_occurrences_of state POD1 = 1 before mark_x***

***bits_0_thru_3*** ***bits_0_3***

Used with the *threshold* command in all modes except Dual Threshold to set the voltage specification against which probe inputs 0-3 will be compared.

**FS -** ***threshold pod_1 bits_0_thru_3 ttl***

***bits_0_thru_3_lower*** ***lower***

Used with the *threshold* command in Dual Threshold Mode to set the voltage specification for the more negative threshold voltage against which to compare probe inputs 0-3.

**FS -** ***threshold pod_1 bits_0_thru_3_lower 0.8 volts and bits_0_thru_3 upper 2.0 volts***

***bits_0_thru_3_upper*** ***upper***

Used with the *threshold* command in Dual Threshold Mode to set the voltage specification for the more positive threshold voltage against which to compare probe inputs 0-3.

**FS -** ***threshold pod_1 bits_0_thru_3_lower 0.8 volts and bits_0_thru_3 upper 2.0 volts***

***bits_0_thru_3_upper_and_lower*** ***u_and_l***

Used with the *threshold* command in Dual Threshold Mode to set the voltage specification for both threshold voltages against which to compare inputs 0-3 to either ttl or ecl. These set the upper and lower thresholds to +2.0V and +0.8V, and -1.1V and -1.5V, for ttl and ecl, respectively.

**FS -** ***threshold pod_1 bits_0_thru_3_upper_and_lower ttl***

***bits_4_thru_7*** ***bits_4_7***

Used with the *threshold* command in Wide Sample Mode to set the voltage specification against which probe inputs 4-7 will be compared.

**FS -** ***threshold pod_1 bits_4_thru_7 ttl***

***blank*** ***blank***

Used with the *display* command to produce an unlabeled null trace in the timing/state diagram, typically used between traces for better readability.

**SD, TD -** ***display POD1.0 then blank then POD1.1***

***bnc_port_4*** ***bnc_4***

Used with the *assert* command to define the conditions for driving rear panel BNC connector #4.

**TS -** ***assert bnc_port_4 on_internal_nondelayed_trigger***

***center_of_trace*** ***center***

Used with the *trigger position_is* command to locate the relative position of the trigger in the center of the acquisition memory, and in X1 magnification, to the center of the displayed trace.

**TS -** ***trigger position_is center_of_trace***

***clock_is*** ***clock***

Used in the External Clock Mode (State Analysis) to set the clock edge and the clock qualifier.

**FS -** ***clock_is rising_edge and high_level_qualify***

***clock_pod*** ***clock_pod***

Used with the *threshold* command to set the threshold level of the clock pod when using state analysis.

**FS -** ***threshold clock_pod ecl***

***compare_faults*** ***compare***

Used with the *process_for_data* command to cause the trace list to show only the compare faults found after comparing bits. This command will appear only after a compare has been done.

**PP -** ***process_for_data compare_faults***

***compare_file*** ***compare***

Used with the *display* command to set up a display which includes information from your compare file.

**SD, TD, TL -** ***display POD1 .6 then compare_file POD1 .6***

***compare_file_is*** ***compare***

Command used to identify a configuration that was stored with data to be accessed by the analyzer and used as a reference source when you are making comparison measurements, or when cancelling an existing compare file specification.

**PP -** ***compare_file_is FILE4***

***configuration*** ***configure***

Command used to create a file of type trace which contains the current Trace, Format, and Post_process Specifications and Display formats for future recall, along with the content of the trace memory (if desired), or to load such a file to set up these specifications again. Should the current analyzer board assembly configuration not be identical to that from which the file was created, labels which will be invalid are shown as null entries in the Format Specification and on execution, an error message will prompt for invalid Trace Specification entries.

**SD, TS, FS, TD, TL, PP -** ***configuration save_in FILENAME:USERID***

***consecutive_faults_allowed_is*** ***faults***

Used with the *compare* command in the Post_process specification to define the number of compare faults allowed. This command requires that a compare file be specified.

**PP -** ***compare faults <samples>***

***copy*** ***copy***

Command used to transfer the Trace, Format, Post_process Specifications, Trace List, and/or Timing/State Diagram to a destination file or device.

**SD, TS, FS, PP, TD, TL -** ***copy all_specifications to printer***

***cursor*** ***cursor***

Command used to cause the **<--** and **-->** keys to operate on the vertical cursor in the diagram rather than on the flashing underline cursor on the command line. The *cursor* softkey is shown in inverse video while this mode is in effect. It can be cancelled by pressing the *cursor* softkey again. Because the command line cursor is eliminated, keyboard entries are inhibited. This command operation is analogous to the revise and insert commands in the Editor.

**SD, TD, TL -** ***cursor***

***date_and_time*** ***date_time***

Used with the *indicate* command to turn the date and time indicator on or off. This is helpful in determining the dates of saved configurations.

**TL, TD, SD -** ***indicate date_and_time on***

***default*** ***default***

Command used to return specifications to their preset conditions. Default conditions for a 16-channel analyzer are:

(Trace) sample rate ***200MHz***;
trigger on ***anything***;
(Format) define ***POD1***, ***POD1upper***, ***POD1lower***,
***POD2***, ***POD2upper***, ***POD2lower***;
threshold ***all_pods all_bits ttl***;
(Diagram) display ***POD1 then POD2 then default_bits_in pod_1 then default_bits_in pod_2*** ;
(Trace List) display ***POD1 in_binary then POD2 in_binary then time_count absolute*** .

**TS, FS, PP -** ***default all_specifications***
**TS, FS, PP -** ***default trace_specification***

***default_bits_in*** ***default***

Used with the *display* command to produce traces labeled as pod_#_bit_#.

**SD, TD -** ***display POD1 then default_bits_in pod_1***

***define*** ***define***

Command used to associate a new label name with probe inputs and a logic polarity. Up to 31 labels can be defined in the Format Specification. Default labels for a 16-channel analyzer are POD1, POD1upper, POD1lower, POD2, POD2upper, and POD2lower.

**FS -** ***define NEWLABEL pod_1_bit 1 logic_polarity positive_true***

***delay_clocks_after*** ***clk_after***

Used with the *trigger* command to delay the trigger position by an integral number of positive clock edges received on the Delay Clock line of the Intermodule Bus, usually driven by the clock in the system under test as perceived by a synchronous analyzer module. A maximum delay of 32767 clocks is possible.

**TS -** ***trigger 5 delay_clocks_after entering POD1 = 0***

***delete*** ***delete***

Command used to remove a label from the Format Specification. If this label is in use in any specification, the command will be ignored.

**FS -** ***delete OLDLABEL***

***diagram*** ***diagram***

Command used to select either 8-channel or 16-channel vertical display format.

**SD, TD -** ***diagram eight_channels***

***display*** ***display***

Command used to set up the sequence of traces to be shown. A maximum of 32 traces can be set up in the Timing/State Diagram, although only 16 can be shown at one time. A maximum width of 132 characters can be set up in the Trace List, although only the first 80 characters will appear on screen. In the timing diagram, the display key used alone will switch between labels and probe bit identifiers for each trace on screen. The default for an eight-channel analyzer is:

**SD, TD -** ***display POD1 then default_bits_in POD1***

**TL -** ***display POD1 then time_count absolute***

***driven*** ***driven***

Used with the *trigger* command to define that the analyzer trigger is to be output to other modules via the High Trigger or Low Trigger Enable lines in the Intermodule Bus.

**TS -** ***trigger driven on POD1 = 0***

**TS -** ***trigger enable driven on_trigger***

***dual_threshold*** ***dual***

Used with the *mode_is* command to set the operating mode of the analyzer to compare each of the 0..3 inputs on each probe pod to two different thresholds and display the result as a three-level trace. If the thresholds have been set previously in this mode, they will return to those values. Otherwise, if the thresholds were set to a voltage value in the previous mode, they will be shifted from that value(s) by +200 mV and -200 mV for the upper and lower thresholds, respectively. If the thresholds were set to ecl, they will go to -1.1V and -1.5V. If the thresholds were set to ttl, or were not set, they will go to +2.0V and +0.8V.

**TS, FS -** ***mode_is dual_threshold***

***ecl*** ***ecl***

Used with the *threshold* command to set the probe input comparison voltage to the middle of the 10K series ECL voltage range, -1.3V. In Dual Threshold Mode, the upper and lower thresholds will be set to -1.1V and -1.5V, respectively.

**FS -** ***threshold pod_1 all_bits ecl***

***eight_channels*** ***8_chanls***

Used with the *diagram* command to set the vertical display format to eight traces.

**SD, TD -** ***diagram eight_channels***

***enable*** ***enable***

Used with the *trigger* command to qualify triggers for or from other modules via the Trigger Enable line on the Inter Module Bus. Note that Trigger Enable and Trigger cannot both be received.

**TS -** ***trigger enable received***

***end*** ***end***

Command used to exit the analyzer software and return to the next higher level, either Measurement_System or system monitor, and to save a copy of the current configuration in a system-defined trace file to be used for automatic re-entry when transferring between modules in a multimodule measurement or when the continue option is used to re-enter either *timing* or *meas_sys*. The trace file name will be Tdc#$:HP where # is the slot number of the Timing Control card and $ is the HP-IB address of the frame if in a cluster environment or 8 if in stand-alone mode. Used with the *copy* command to indicate the range of the copy is from the present cursor position to the end of the trace memory.

**SD, TS, FS, PP, TD, TL -** ***end***

**TL -** ***copy trace_list thru end***

***end_of_trace*** ***end***

Used with the *trigger position_is* command to define the position of the trigger within the acquisition memory as 30 samples before the end of memory (70 in Fast Sample Mode).

**TS -** ***trigger position_is end_of_trace***

*entering* *entering*

Used with the *trigger* command to define the trigger condition as the point where the trigger pattern comparison changes from not-true to true. The not-true condition must precede the true; if the condition is always true, triggering will not occur. If the trigger is output to the rear panel BNC connectors, the signal there will be a narrow pulse. Note that an ANDed condition will go true when the total condition goes true, e.g.- all elements are true but one and then that element goes true, causing the trigger. Used with the *find* and *mark* commands to define the point to be found (and/or marked) as the point where a specified pattern changes from not-true to true. The not-true condition must precede the true; if the condition is always true, it will not be found.

**TS -** ***trigger on entering POD1. 0 = 0 and POD1. 1 = 1***
(note that ***trigger on entering POD1 = 0XXH*** will never trigger)

**SD, TD, TL -** ***find entering POD1 .6 = 1 and POD1 .5 = 0 thru mark_x***

**SD, TD, TL, PP -** ***mark a on_all_occurrences_of entering POD1 .6 = 1 and POD1 .7 = 1***

*equal* *equal*

Used with the *halt_repetitive_execution* command to cause the analyzer to halt the current measurement when the bits being compared are equal.

**PP -** ***halt_repetitive_execution when_compare equal***

*execute* *execute*

Command to cause the analyzer to start capturing data. If any Trace Specifications have changed since the last measurement, the new setup will be loaded before capturing data. When the defined pretrigger data has been stored, the analyzer will then begin checking for its trigger condition which will be used to determine when the measurement is complete.

**SD, TS, FS, PP, TD, TL -** ***execute***

*external_clock* *ext_clock*

Used with the *mode* command to set up the analyzer for state analysis.

**FS, TS -** ***mode_is external_clock***

*falling_edge* *falling*

Used with the *clock* command in the External Clock Mode to set the clock edge.

**FS -** ***clock_is falling_edge and high_level_qualify***

*fast_sample* *fast*

Used with the *mode_is* command to set the operating mode of the analyzer to sample each of the 0..3 inputs on each probe pod at a 400-MHz rate with a memory depth of 8140 samples.

**SD, TS, FS -** ***mode_is fast_threshold***

*find* *find*

Command to bring any specified event to the display, centered if possible, and locate the cursor at that point. Using this command, you can find the trigger, or any of the mark_x, o, a, b, c, or d samples by specifying the mark_identifiers. You can also find types of occurrences such as when a probe channel first enters or leaves a specified value, or when a condition lasts for a time greater than or less than a specified time.

**SD, TD, TL -** ***find mark_x***

***followed_by*** ***followed***

Used with the *trigger* command to establish a trigger sequence composed of two conditions. The second condition can be recognized only after the first condition has been satisfied. Note that by this definition, a second condition which is true when the first condition becomes true will produce a trigger. No ambiguity can occur if you use the *entering* or *leaving* constructs to define your second condition.

**TS -** ***trigger on entering POD1 = 0 followed_by entering POD2 = 0***

***format_specification*** ***format***

Used with the *show* command to bring the Format Specification on the screen for setting labels, thresholds, and logic polarities.

**TS, PP, TD, TL -** ***show format_specification***

***from*** ***from***

Used with the *compare* command to define a sample range in the trace memory for the compare operation to occur. This command requires that a compare file be specified.

**PP -** ***compare all_bits from cursor thru end***

***glitch_capture*** ***glitch***

Used with *mode_is* command to set the operating mode of the analyzer to examine each of the 0..3 inputs on each probe pod for both data and glitches. A glitch is defined as multiple transitions between adjacent data samples. Triggers involving glitches are synchronous with durations in multiples of sample periods. A Glitch is displayed at x100, x40 and x20 magnification as adjacent broken vertical bars. At x10, x4, x2 and X1, a single broken bar is used to identify both glitches and multiple data transitions occurring too near each other to be displayed separately.

**TS, FS -** ***mode_is glitch_capture***

***graphic*** ***graphic***

Used with the *copy* command to produce a graphic illustration of the timing/state diagram. This option is only available when a graphic screen printer card is installed in the development station.

**SD, TD -** ***copy diagram graphic to printer***

***greater_than*** ***greater***

Used with the *trigger* command to time-qualify the trigger pattern, allowing for trigger recognition only on those patterns which exist for a time greater than the specification. The trigger position will indicate where the time specification was satisfied. If the trigger is output to the rear panel BNC connectors, the signal there will be that portion of the trigger pattern that exceeds the time specification. The available range is from 5 nsec to 1 msec for asynchronous triggers and from two sample periods min to 1 msec max for synchronous triggers. Used with the *process* command to qualify the display of only those samples which exceeded the time specification. Used with the *find* command to define the point to be found as the point where a specified pattern remains true for more than a particular time period. The analyzer can only find the specified condition within the range of memory you designate. Used with the *halt_repetitive_execution* command to designate a number of marks or a period of time which, when exceeded, will cause generation of the halt command for a repetitive execution in progress.

**TS -** ***trigger on greater_than 50 nsec_of POD1 = 0***
**TL, PP -** ***process_for_data greater_than 30 usec on POD1 .6***
**TD, TL -** ***find greater_than 30 usec_of POD1 .6 = 1 and POD1 .5 = 0 thru mark_x***
**PP -** ***halt_repetitive_execution when_marks_x_o greater_than 40***

***halt*** ***halt***

Command available only while executing, to stop the current execution. If memory has been filled, the entire memory can be displayed, otherwise only that portion of memory just acquired can be shown. The trigger is placed at the end of memory.

**SD, TS, FS, TD, TL, PP -** ***halt***

***halt_repetitive_execution*** ***halt_rept***

Command used to increase resolution of duration triggers, increase sequencing capability, and to provide event-count triggering. Execution finishes when your specified condition is found.

**PP -** ***halt_repetitive_execution when_time_x_o greater_than 40 usec***

***hertz*** ***Hz***

Used with the *sample* command to set the frequency multiplier of the analyzer sample clock.

**TS, TD, TL -** ***sample rate_is 2 hertz***

***high_level_qualify*** ***high_qual***

Used with the *clock* command in the External Clock Mode to set the clock qualifier.

**FS -** ***clock_is rising_edge and high_level_qualify***

***hpib*** ***hpib***

Used with the *copy* command to copy specifications to the hpib port. This command is also used with the *configuration* command to load configurations from or save configurations to the hpib port. This option is only available when the analyzer is used in a stand alone development station.

**TS, FS, TD, SD, PP, TL -** ***copy trace_specification to hpib***
**TS, FS, TD, SD, PP, TL -** ***configuration load from hpib***

***indicate*** ***indicate***

Command used to call for display of the time interval measured between the mark_x and mark_o samples, or to indicate the number of mark_<abcd> samples between the mark_x and mark_o samples. Additionally, in the Timing/State Diagram, the command can be used to call for display of the state of each trace at the cursor position. The arrowheads at the end of each trace label are replaced with h, l, m, or g to show high, low, middle, or glitch. This command is also used to display the date and time in the timing/state diagram and tracelist.

**SD, TD -** ***indicate levels_at_cursor***
**TD, TL -** ***indicate time_interval_x_o***
**SD -** ***indicate states_x_o***
**TD, SD, TL -** ***indicate date_and_time***

***indicator*** ***indicator***

Used with the *magnify* command to turn on or off the horizontal bar, above each trace, which shows the region that will be displayed at the next higher level of magnification.

**SD, TD -** ***magnify indicator on***

***in_bin*** ***in_bin***

Used with the *display* command to define that the preceding label will be displayed in binary format.

**TL -** ***display POD1 in_bin***

***in_dec*** ***in_dec***

Used with the *display* command to define that the preceding label will be displayed in decimal format.

**TL -** ***display POD1 in_dec***

***in_hex*** ***in_hex***

Used with the *display* command to define that the preceding label will be displayed in hexadecimal format.

**TL -** ***display POD1 in_hex***

***in_oct*** ***in_oct***

Used with the *display* command to define that the preceding label will be displayed in octal format.

**TL -** ***display POD1 in_oct***

***kilohertz*** ***KHz***

Used with the *sample* command to set the frequency multiplier of the analyzer sample clock.

**TS, TD, TL -** ***sample rate_is 20 kilohertz***

*leaving* *leaving*

Used with the *trigger* command to define the trigger as the point where a particular pattern changes from true to not-true. The true condition must precede the not-true; if the condition is always not-true, triggering will not occur. If the trigger drives the rear panel BNC connector, the signal there will be a narrow pulse. Note that an ANDed condition will trigger when the entire condition goes false, e.g.- all elements are true and then at least one element goes false to cause the trigger. Used with the *find* and *mark* commands to define the point to be found (and/or marked) as the point where a particular pattern changes from true to not-true. The true condition must precede the not-true; if the condition is always not-true, the event will not be found (or marked).

**TS -** ***trigger on leaving POD1 .0 = 0 and POD1 .1 = 0***
(note that ***trigger on leaving POD1 = 0XXH*** will never trigger)

**SD, TD, TL -** ***find leaving POD1 .6 = 1 and POD1 .5 = 0 thru mark_x***

**TD, TL, PP -** ***mark a on_all_occurrences_of POD1 .0 = 1***

**SD -** ***mark a on_all_occurrences_of leaving POD1 = 0***

*less_than* *less*

Used with the *trigger* command to time qualify the trigger pattern, allowing triggers to be only those patterns which exist for a time shorter than the specification. The trigger position will indicate where the time specification was satisfied. If the trigger is output to the rear panel BNC connector, the signal there will be a narrow pulse at the time that the pattern goes false. The available time range is from 20 nsec to 1 msec for asynchronous triggers, and from two sample periods minimum to 1 msec maximum for synchronous triggers. Used with the *find* command to define the point to be found as the point where a particular pattern remains true for less than the specified time period. The analyzer can only find the specified condition within the range of memory you designate. Used with the *halt_repetitive_execution* command to designate a number of marks or a period of time which, when not met, will cause generation of the halt command for the repetitive execution in progress.

**TS -** ***trigger on less_than 50 nsec_of POD1 = 0***

**TD, TL -** ***find less_than 30 nsec_of POD1 .6 = 1 and POD1 .5 = 0 thru mark_x***

**PP -** ***halt_repetitive_execution when_marks_x_o less_than 40***

*levels_at_cursor* *levels*

Used with the *indicate* command to display the state of each trace at the cursor position. The arrowheads beside each trace label are replaced with h, l, m or g to show high, low, middle or glitch.

**SD, TD -** ***indicate levels_at_cursor***

*load_from* *load_from*

Used with the *configure* command to restore the Trace, Format, and Post_process Specifications and Timing/State Diagram and Trace List formats to those contained in a trace file saved previously. The data obtained during a previous measurement can also be restored to the trace memory if it was stored along with the configuration.

**SD, TS, FS, TD, TL, PP -** ***configuration load_from FILENAME:USERID***

*log_to_file* *log_file*

Used with the *stats* command to log the statistics to a file after each execution.

**PP -** ***statistics log_to_file <FILENAME>***

*logic_polarity* *polarity*

Used with the *threshold* command to set the sense of a "1" as more positive or more negative than the probe threshold voltage for positive-true or negative-true, respectively. This allows trigger definitions to be made in terms of "1's" and "0's" independent of the volt-sense of the lines being measured. Trace List values will reflect these definitions but the Timing/State Diagram will show only positive-true voltage comparisons in oscilloscope fashion to prevent ambiguity.

**FS -** ***threshold pod_1 all_bits ttl logic_polarity negative_true***

*low_level_qualify* *low_qual*

Used with the *clock* command in the External Clock Mode to set the clock qualifier.

**FS -** ***clock_is rising_edge and low_level_qualify***

*magnify* *magnify*

Command used to change the time per division of the horizonal axis, or to turn on or off the bar which indicates the region that can be viewed at the next x10 level of magnification.

**SD, TD -** ***magnify x10***

*mark* *mark*

Command used to assign identifiers to samples in the trace memory for use in making measurements between samples and identifying events in trace memory.

**SD, TD, TL, PP -** ***mark x***

*marked* *marked*

Used with the *process* command to bring only the marked samples to the screen.

**TL -** ***process_for_data marked***

*mark_a* *mark_a*

Used with the *find* command to bring any location marked with *a* to the display, centered if possible, and place the cursor on it (*mark_b*, *mark_c*, and *mark_d* work the same as *mark_a*). Used with the *halt_repetitive_execution* command to designate a member of a sequence of occurrences which, when found in order, will halt the repetitive executions in progress.

**SD, TD, TL -** ***find mark_a***

**PP -** ***halt_repetitive_executions when_sequence_x_o***
***mark_a then mark_a then_not mark_c***

*mark_b* *mark_b*

Refer to ***mark_a*** .

*mark_c* *mark_c*

Refer to ***mark_a*** .

*mark_d* *mark_d*

Refer to ***mark_a*** .

*mark_names* *mark_name*

Used with the *display* command to get a column on screen that shows the mark names you assigned in the Post_process Specification.

**TL -** ***display POD1 in_hex then mark_names***

*mark_o* *mark_o*

Used with the *find* command to bring the location marked with *o* to the display, centered if possible, and put the cursor on that sample.

**SD, TD, TL -** ***find mark_o***

*mark_x* *mark_x*

Used with the *find* command to bring the location marked with *x* to the display, centered if possible, and put the cursor on that sample.

**SD, TD, TL -** ***find mark_x***

*maximum_and_minimum* *max_min*

Used with the *indicate* command to obtain two additional counts on the display, one indicating the largest number and the other indicating the smallest number ever measured during the series of repetitive executions of the selected measurement.

**SD, TD, TL -** ***indicate number_of_marks_x_o maximum_and_minimum***

*mean_and_standard_deviation* *mean_stdv*

Used with the *indicate* command to obtain two additional counts on the display relating to repetitive executions of a selected measurement: the *mean* number indicates the average count obtained during the measurement, and the *stdv* number indicates the standard deviation from the mean value.

**SD, TD, TL -** ***indicate number_of_marks_x_o mean_and_standard_deviation***

*measurement_data_in_binary* *meas_data*

Used with the *copy* command to copy the measurement data to a file of type :data.

**SD, TD, TL -** ***copy measurement_data_in_binary to <FILE>***

*megahertz* *MHz*

Used with the *sample* command to set the frequency multiplier of the analyzer sample clock.

**TS -** ***sample rate_is 200 megahertz***

*mode_is* *mode*

Command used to change the operating modes of the analyzer to Dual Threshold, Fast Sample, Glitch Capture, Wide Sample or External Clock Mode. In Dual Threshold Mode, the lower numbered four inputs on each probe pod are compared to two voltages and displayed as three-level traces. The Fast Sample mode samples the lower four inputs at a 400-MHz rate. In Glitch Capture Mode, the occurrence of multiple data transitions or glitches on the lower inputs is recorded. Wide Sample Mode uses all eight probe inputs for maximum input width. External Clock Mode connects an external clock source to the analyzer to provide the system clock.

**TS, FS -** ***mode_is dual_threshold***

***modify*** ***modify***

Used with the *trigger*, *trigger enable*, *display*, and *process* commands to recall the current specification to the command line for editing. Using this command avoids extensive re-entry typing.

**TS -** ***trigger modify***
**TS -** ***trigger enable modify***
**SD, TD, TL -** ***display modify***
**TL, PP -** ***process_for_data modify***
**FS -** ***modify LABEL***

***msec*** ***msec***

Used with the *sample* command to set the period multiplier of the analyzer sample clock. Used with the *halt_repetitive_execution* command to identify the time period to be referenced when the analyzer determines whether or not to stop repetitive executions in progress.

**TS, TD, TL -** ***sample period_is 25 msec***
**PP -** ***halt_repetitive_executions when_time_x_o greater_than 35 msec***

***msec_after*** ***ms_after***

Used with the *trigger* command to enter a time multiplier specifying the delay from when the trigger pattern definition is satisfied to when the trigger occurs. The delay range is from 2 to 32,000,000 sample periods, or from 1 to 32767 msec, whichever is less.

**TS -** ***trigger 123 msec_after POD1 = 0***

***msec_of*** ***msec_of***

Used with the *trigger* command to enter the multiplier for time qualification on trigger patterns.

**TS -** ***trigger on greater_than 1 msec_of POD1 = 0***

***negative_true*** ***negative***

Used with the *define* command to specify that for this label, a voltage more negative than the threshold voltage will be considered a "1" in the Trace Specification *trigger* definition and in the Trace List.

**FS -** ***threshold pod_1 all_bits ttl logic_polarity negative_true***

***neg_transition_on*** ***negative***

Used with the *process* command to specify that a transition on the selected probe line will serve as the reference event.

**TL, PP -** ***process_for_data sampled 3 samples_after neg_transition_on POD1 .6***

***none*** ***none***

Used with the *compare_file_is* command to cancel an existing compare file specification.

**PP -** ***compare_file_is none***

***not_equal*** ***not_equal***

Used with the *halt_repetitive_execution* command to cause the analyzer to halt the current measurement when the bits being compared are not equal. This command requires that a compare file be specified.

**PP -** ***halt_repetitive_execution when_compare not_equal***

***nsec*** ***nsec***

Used with the *sample* command to set the period multiplier of the analyzer sample clock. Used with the *halt_repetitive_execution* command to identify the time period to be referenced when the analyzer determines whether or not to stop repetitive executions in progress.

**TS, TD, TL -** ***sample period_is 25 nsec***

**PP -** ***halt_repetitive_executions when_time_x_o greater_than 200 nsec***

***nsec_after*** ***ns_after***

Used with the *trigger* command to enter a time multiplier specifying the delay from when the trigger pattern definition is satisfied to when the trigger occurs. The delay range is from 2 to 32,000,000 sample periods, or from 1 to 32767 nsec, whichever is less.

**TS -** ***trigger 100 nsec_after POD1 = 0***

***nsec_of*** ***nsec_of***

Used with the *trigger* command to enter the multiplier for time qualification on trigger patterns.

**TS -** ***trigger on greater_than 100 nsec_of POD1 = 0***

***number_of_marks_x_o*** ***marks***

Used with the *indicate* command to obtain a count on the display of the number of mark_<abcd> samples between the mark_x sample and the mark_o sample in memory.

**SD, TD, TL -** ***indicate number_of_marks_x_o***

***o*** ***o***

Used with the *mark* command to identify a particular sample in trace memory for measurements between samples or for ease of finding that sample.

**SD, TD, TL -** ***mark o***

***off*** ***off***

Used with the *magnify*, *indicate*, and *mark* commands to turn off the magnification indicator bar or level indicator characters, and to turn marks *on* and *off*. Used with the *process* command to turn off display qualification, allowing display of all samples in memory. Used with the *halt_repetitive_execution* command to specify that repetitive executions of the measurement will not be halted.

**SD, TD -** ***magnify indicator off***

**SD, TD -** ***indicate levels_at_cursor off***

**TD, TL -** ***mark x off***

**TL, PP -** ***process_for_data off***

**PP -** ***halt_repetitive_executions off***

**TD, SD, TL -** ***indicate date_and_time off***

***on*** ***on***

Used with the *trigger* command to define a pattern to be used for triggering, with *magnify* and *indicate* commands to enable the magnification indicator bar and level indicator characters, and with the *mark* command to enable the marks.

**TS -** ***trigger on POD1 = 4AH***
**SD, TD -** ***magnify indicator on***
**SD, TD -** ***indicate levels_at_cursor on***
**TD, TL, PP -** ***mark x on***
**TD, SD, TL -** ***indicate date_and_time on***

***on_all_occurrences_of*** ***on_all***

Used with *mark <abcd>* to specify events to be identified with the corresponding marks.

**SD, TD, TL, PP -** ***mark a on_all_occurrences_of entering POD1 .6 = 1 after trigger***

***on_cursor*** ***on_cursor***

Used with the *mark x* and *mark o* commands to assign a mark to follow the cursor position wherever it moves.

**SD, TD, TL, PP -** ***mark o on_cursor***

***on_first_occurrence_of*** ***on_first***

Used with the *mark x* and *mark o* commands to define the event to be identified with the corresponding mark.

**SD, TD, TL, PP -** ***mark x on_first_occurrence_of entering POD1 .6 = 1 after trigger***

***on_internal_nondelayed_trigger*** ***on_triggr***

Used with the *assert* command to define that the BNC port will be driven when the trigger condition occurs. Any delay time, samples or IMB clock delays are not applied.

**TS -** ***assert bnc_port_4 on_internal_nondelayed_trigger***

***on_sample*** ***on_sample***

Used with the *mark_x* and *mark_o* commands to place an identifying mark on a numbered sample. The trigger position is defined as sample 0 and all samples are numbered + or - with respect to the trigger location.

**SD, TD, TL, PP -** ***mark o on_sample 256***

***on_trigger*** ***on_triggr***

Used with the *trigger* command to define when the Low Trigger Enable line in the Inter Module Bus will be driven. Used with the *mark x* and *mark o* commands to place a mark on the trigger position.

**TS -** ***trigger enable driven on_trigger***
**SD, TD, TL, PP -** ***mark x on_trigger***

***or*** ***or***

Used with the *trigger* command (using greater than 16 channels) to specify alternate conditions for the trigger condition.

**TS -** ***trigger on entering POD1 = 0 or POD2 = 0***

***or_on*** ***or_on***

Used with the *trigger* command in Glitch Mode to define additional inputs on which glitches can be recognized as triggers.

**TS -** ***trigger on any_glitch on POD1. 0 or_on POD1. 1***

***percent_after_start_of_trace*** ***after***

Used with the *trigger* command to locate the relative position of the trigger within the acquisition memory in 5% increments after the beginning of memory.

**TS -** ***trigger position_is 20***
***percent_after_start_of_trace***

***percent_before_end_of_trace*** ***before***

Used with the *trigger* command to locate the relative position of the trigger within the acquisition memory in 5% increments before the end of memory.

**TS -** ***trigger position_is 20***
***percent_before_end_of_trace***

***period_is*** ***period***

Used with the *sample* command to define the time interval between samples. The allowed range is 5 nsec (10 nsec in Glitch Mode) to 500 msec.

**TS, TD, TL -** ***sample period_is 50 nsec***

***pod_1*** ***pod_1***

Used with the *threshold* command to define the pod to which the threshold voltage definition applies. Used with the *display* command to format the sequence of traces on screen with labels showing probe inputs. The probe connected to the Acquisition board in the lower numbered slot for a 16-channel analyzer is defined as probe pod #1. In an 8-channel analyzer, it is the only probe pod. While the POD1 label is redefinable, *pod_1* is not.

**FS -** ***threshold pod_1 all_bits ecl***
**SD, TD, TL -** ***display default_bits_in pod_1***

***pod_1_bit*** ***pod_1_bit***

Used with the *define* command to establish a mapping between a label and probe pod #1 inputs.

**FS -** ***define NEWLABEL pod_1_bit 0 width 3***

***pod_2*** ***pod_2***

Used with the *threshold* command to define the pod to which a threshold voltage applies. Used with the *display* command to format the sequence of traces on screen with labels showing probe inputs. The probe that connects to the Acquisition Board in the higher numbered slot for a 16-channel analyzer is defined as probe pod #2. In an 8-channel analyzer, *pod_2* is undefined. While the label POD2 is redefinable, *pod_2* is not.

**FS -** ***threshold pod_2 all_bits ecl***
**SD, TD, TL -** ***display default_bits_in pod_2***

***pod_2_bit*** ***pod_2_bit***

Used with the *define* command to establish a mapping between a label and the inputs of probe pod #2 in a 16-channel analyzer.

**FS -** ***define NEWLABEL pod_2_bit 0 thru pod_1_bit 6***

***pod_3*** ***pod_3***
Used with a 24-channel analysis subsystem. See ***pod_2*** definition.

***pod_3_bit*** ***pod_3_bit***
Used with a 24-channel analysis subsystem. See ***pod_2_bit*** definition.

***pod_4*** ***pod_4***
Used with a 32-channel analysis subsystem. See ***pod_2*** definition.

***pod_4_bit*** ***pod_4_bit***
Used with a 32-channel analysis subsystem. See ***pod_2_bit*** definition.

***position_is*** ***position***
Used with the *trigger* command to locate the relative position of the trigger within the trace memory.
**TS -** ***trigger position_is start_of_trace***

***positive_true*** ***positive***
Used with the *define* command to specify that for this label, a voltage more positive than the threshold voltage will be considered a "1" for trigger definition in the Trace Specification, and in the Trace List.
**FS -** ***threshold pod_1 all_bits ttl logic_polarity positive_true***

***post_process_specification*** ***postspec***
Used with the *show* command to select display of the Post_process Specification.
**SD, TS, FS, TL, TD -** ***show post_process_specification***

***pos_or_neg_transition_on*** ***pos_neg***
Used with the *process* command to specify that either a positive-going or negative-going transition on the selected probe line will serve as the reference event.
**TL, PP -** ***process_for_data sampled 3 samples_after pos_or_neg_transition_on POD1 .6***

***pos_transition_on*** ***positive***
Used with the *process* command to specify that a positive-going transition on the selected probe line will serve as the reference event.
**TL, PP -** ***process_for_data sampled 2 samples_before pos_transition_on POD1 .7***

***printer*** ***printer***
Used with the *copy* command to define the printer as the destination device to which copies of Specifications will be sent.
**SD, TS, FS, PP, TD, TL -** ***copy all_specifications to printer***

***process_for_data*** ***process***
Command used for display qualification. It allows only samples that meet your specification to be displayed on screen.
**PP, TL -** ***process_for_data sampled 2 samples_before pos_transition_on POD1 .6***

*rate_is* *rate*

Used with the *sample* command to define the frequency at which samples are taken. The allowed range is from 2 Hz to 200 MHz (100 MHz in Glitch Mode).

**TS, TD, TL -** ***sample rate_is 200 megahertz***

*received* *received*

Used with the *trigger* command to specify the Trigger or Trigger Enable line in the Intermodule Bus as the trigger source, or for enabling the internal trigger, respectively. The trigger and trigger enable cannot be received simultaneously.

**TS -** ***trigger received***

**TS -** ***trigger enable received***

*received_or_driven* *rec/drive*

Used with the *trigger* command to receive the Trigger line in the Intermodule bus as a trigger source, or if the internal trigger condition is satisfied first, to trigger and also drive this line (this is the logical OR of an internal and an external trigger).

**TS -** ***trigger received_or_driven***

*relative* *relative*

Used with the *display* command to define that the time displayed will be differential time between the samples on each line.

**TL -** ***display POD1 then time_count relative***

*rename* *rename*

Command used to alter an existing label by changing its name.

**FS -** ***rename OLDLABEL to NEWLABEL***

*repetitively* *repeat*

Used with the *execute* command to require that another measurement begin as soon as the current measurement is complete. Measurements will continue until the *halt* key is pressed.

**SD, TS, FS, PP, TD, TL -** ***execute repetitively***

*restart* *restart*

Used with the *trigger* command to enable control of analyzer measurements by another module on the Intermodule Bus. This option is only available after *trigger enable received* has been specified. When Trigger Enable goes false, the analyzer resets and begins a new measurement; when Trigger is true, the measurement in progress is allowed to complete, or the data in memory is retained.

**TS -** ***trigger enable restart***

*rewritten_with_current_measurement* *rewrite*

Used with the *compare* command to indicate that the existing compare file is rewritten with the current measurement in the analyzer. This command requires that a compare file be specified.

**PP -** ***compare file_is rewrite***

*rising_edge* *rising*

Used with the *clock* command in the External Clock Mode to set the clock edge.

**FS -** ***clock_is rising_edge***

***sample*** ***sample***

Command used to set the frequency or period at which data will be acquired and stored by the analyzer. The allowed range is from 2 Hz/500 msec to 200 MHz/5 nsec (2 Hz/500 msec to 100 MHz/10 nsec in Glitch Mode). The default is the fastest available sample rate in each mode.

**TS, TD, TL -** ***sample rate_is 200 megahertz***

***sampled*** ***sampled***

Used with the *process* command for display qualification. This qualification allows display of only those samples that occurred at the specified locations.

**TL, PP -** ***process_for_data sampled 2 samples_after neg_transition_on POD1 .6***

***samples_after*** ***after***

Used with the *process* command to qualify display of the selected sample after the designated occurrence.

**TL, PP -** ***process_for_data sampled 2 samples_after neg_transition_on POD1 .6***

***samples_before*** ***before***

Used with the *process* command to qualify display of the selected sample before the designated occurrence.

**TL, PP -** ***process_for_data sampled 2 samples_before neg_transition_on POD1 .2***

***samples_of*** ***samples***

Used with the *process_for_data* command to cause the trace list to show only the samples defined.

**PP, TL -** ***process_for_data samples_of POD1 = 0***

***save_in*** ***save_in***

Used with the *configure* command to store the current Trace, Format, and Post_process Specifications and the Timing/State Diagram and Trace List formats (and trace memory content, if desired), in a trace file.

**SD, TS, FS, PP, TD, TL -** ***configuration save_in FILENAME:USERID with_data***

***sec*** ***sec***

Used with the *halt_repetitive_execution* command to identify the time period to be referenced when the analyzer determines whether or not to stop the repetitive executions in progress.

**PP -** ***halt_repetitive_executions when_time_x_o greater_than 10 sec***

***sec_after*** ***sec_after***

Used with the *trigger* command to enter a time multiplier specifying the delay from when the trigger pattern definition is satisfied to when the trigger occurs. The delay range is from 2 to 32,000,000 sample periods, or 1 to 32767 sec, whichever is less.

**TS -** ***trigger 1 sec_after POD1 = 0***

***show*** ***show***

Command used to change the display to another specification or to the Timing/State Diagram or Trace List.

**SD, TS, FS, TD, PP -** ***show trace_list***
**TS, FS, PP, TL -** ***show timing_diagram***
**SD, TS, PP, TD, TL -** ***show format_specification***
**SD, PP, FS, TD, TL -** ***show trace_specification***
**SD, TS, FS, TD, TL -** ***show post_process_specification***

***sixteen_channels*** ***16_chanls***

Used with the *diagram* command to set the vertical display format to sixteen traces on screen.

**SD, TD -** ***diagram sixteen_channels***

***start*** ***start***

Used with the *copy* command to indicate that the range of information to be transferred begins at the start of the trace and ends at the present cursor position.

**TL -** ***copy trace_list thru start***

***start_of_trace*** ***start***

Used with the *trigger position_is* command to specify the trigger position is 30 samples after the start of the trace (70 in Fast Sample Mode).

**TS -** ***trigger position_is start_of_trace***

***state*** ***state***

Used with the *trigger* command in the External Clock Mode to trigger on a specified state.

**TS -** ***trigger on state POD1 = 1***

***state_count*** ***state_cnt***

Used with the *display* command to show either the absolute or relative state count.

**TL -** ***display state_count absolute***

***state_diagram*** ***diagram***

Used with the *show* command to bring the State Diagram on screen for viewing the results of a trace or formatting the display sequence.

**TS, FS, PP, TL -** ***show state_diagram***

***states_after*** ***states***

Used with the *trigger* command in the External Clock Mode to delay the trigger a specified number of states.

**TS -** ***trigger <DELAY> states_after state POD1 = 1***

***states_of*** ***states***

Used with the *process* command in the External Clock Mode to cause the trace list to show only the states defined.

**TL -** ***process_for_data states_of POD1 = 1***

***states_x_o*** ***states***

Used with the *indicate* command to display either the maximum and minimum number of states between the x and o marks, or the mean and standard deviation of the states between the x and o marks.

**SD, TL -** ***indicate states_x_o maximum_and_minimum***

***statistics*** ***stats***

Used with the *copy* command to copy the statistics to a file or printer. This command is also used to limit the range over which statistics are to occur, and to log statistics to a file or printer.

**SD, TD, TL -** ***copy statistics to printer***
**PP -** ***stats when_time_x_o less_than 50 nsec***
**PP -** ***stats log_to_file <FILE>***

***then*** ***then***

Used with the *display* command to add another trace when formatting the display sequence. Used with the *halt_repetitive_execution* command to add another event to a sequence of events which must be found to generate the halt command for a series of repetitive executions.

**SD, TD, TL -** ***display POD1.0 then POD1.2***
**PP -** ***halt_repetitive_execution when_sequence_x_o mark_a then mark_b then mark_c***

***then_not*** ***then_not***

Used with the *halt_repetitive_execution* command to include a not-occurrence in a sequence of occurrences which must be found to generate the halt command for a series of repetitive executions.

**PP -** ***halt_repetitive_execution when_sequence_x_o mark_a then_not mark_d then mark_a***

***threshold*** ***threshold***

Command to set the comparison voltage(s) that the analyzer uses to determine logic levels of the probe input signals. The allowed range is from -10.0V to +10.0V in 0.1V steps. Default is ttl (+1.4V).

**FS -** ***threshold all_pods all_bits ecl***

***thru*** ***thru***

Used with the *define* command to specify the highest numbered bit in a range of bits to be included in the definition of a label. Used with the *copy* command to indicate the end of a range of lines to be transferred to a file or device.

**FS -** ***define LABEL pod_1_bit 0 thru pod_1_bit 4***
**SD, TD, TL -** ***copy trace_list thru end to printer***

***time_count*** ***time_cnt***

Used with the *display* command to include a column showing either relative or absolute times.

**TL -** ***display POD1 then time_count relative***

***time_interval_x_o*** ***time_x_o***

Used with the *indicate* command to display the time interval between the mark_x sample and the mark_o sample in memory.

**TD, TL -** ***indicate time_interval_x_o***

***timing_diagram*** ***diagram***

Used with the *show* command to bring the Timing Diagram on screen for viewing the results of a trace or formatting the display sequence.

**TS, FS, PP, TL -** ***show timing_diagram***

***to*** ***to***

Used with the *rename* command to specify a new name to be used in place of an existing label name. Used with the *copy* command to identify the destination file or device to which a specification is to be copied.

**FS -** ***rename OLDLABEL to NEWLABEL***

**SD, TS, FS, PP, TD, TL -** ***copy all_specifications to printer***

***to_compare_file*** ***to***

Used with the *compare* command to define the bits to be compared. This command requires that a compare file be specified.

**PP -** ***compare POD1 . 0 to_compare_file POD1 . 0***

***trace_list*** ***tracelist***

Used with the *show* command to indicate that the trace list will be brought to the display. Used with the *copy* command to indicate that the Trace List Display is to be transferred to a destination file or device.

**TS, FS, PP, TD -** ***copy trace_list***

**TL -** ***copy trace_list to FILENAME:USERID***

**SD -** ***show tracelist***

***trace_specification*** ***tracespec***

Used with the *show* command to bring the Trace Specification onto the screen for setting triggers, sample rate, and interactions with other modules.

**SD, TS, PP, TD, TL -** ***show trace_specification***

***trigger*** ***trigger***

Command to define the pattern(s) of logic levels to be found on the probe inputs, and to include time or event delays, Intermodule Bus signal interactions, and relative memory positioning (which can be used for ending an execution). The default is a "don't-care" trigger. Used with the *find* command to specify that the trigger sample is to be placed at the center of the screen, if possible, and to locate the cursor at that point.

**TS -** ***trigger on any_thing***

**SD, TD, TL -** ***find trigger***

***ttl*** ***ttl***

Used with the *threshold* command to specify the comparison voltage for the probe inputs as the middle of the TTL voltage range, +1.4V. In Dual Threshold Mode, the upper and lower threshholds are set to +2.0V and +0.8V, respectively.

**FS -** ***threshold pod_1 all_bits ttl***

***usec*** ***usec***

Used with the *sample* command to set the period multiplier of the analyzer sample clock. Used with the *halt_repetitive_execution* command to identify the time period to be referenced when the analyzer determines whether or not to stop a series of repetitive executions.

**TS, TD, TL -** ***sample period_is 25 usec***

**PP -** ***halt_repetitive_executions when_time_x_o***
***greater_than 200 usec***

***usec_after*** ***us_after***

Used with the *trigger* command to enter a time multiplier for specifying delay between recognition of the trigger pattern and identification of the trigger sample. The delay range is from 2 to 32,000,000 sample periods, or from 1 to 32767 msec, whichever is less.

**TS -** ***trigger 123 msec_after POD1 = 0***

***usec_of*** ***usec_of***

Used with the *trigger* command to enter a multiplier for time qualification of trigger patterns.

**TS -** ***trigger on greater_than 1 usec_of POD1 = 0***

***volts*** ***volts***

Used with the *threshold* command to enter a voltage value for comparison with the voltage of each probe input when determining signal logic levels.

**FS -** ***threshold pod_1 all_bits 5.0 volts***

***when*** ***when***

Used with the *compare* command to define when the compare operation is to occur. This command requires that a compare file be specified.

**PP -** ***compare all_bits when POD1 = 10H***

***when_compare*** ***compare***

Used with the *halt_repetitive_execution* command to cause the analyzer to halt the current measurement when the bits being compared are equal or not equal.

**PP -** ***halt_repetitive_execution when_compare equal***

***when_marks_x_o*** ***marks***

Used with the *halt_repetitive_execution* command to specify a number of *mark_<abcd>* samples to be counted between the mark_x and mark_o samples. When the count exceeds the specified number of samples, the analyzer will generate the *halt* command for repetitive executions.

**PP -** ***halt_repetitive_execution when_marks_x_o***
***greater_than 26***

***when_runs_equals*** ***runs***

Used with the *halt_repetitive_execution* command to specify a number of runs to be included in a measurement made up of a series of repetitive executions.

**PP -** ***halt_repetitive_execution when_runs_equals 99***

***when_sequence_x_o*** ***sequence***

Used with the *halt_repetitive_executions* command to specify a sequence of occurrences (identified by mark_<abcd> designators) which, when found, will cause the analyzer to generate the *halt* command for repetitive executions.

**PP -** ***halt_repetitive_execution when_sequence_x_o***
***mark_a then mark_b then_not mark_c***

***when_states_x_o*** ***states***

Used with the *halt_rept* and *stats* commands in the External Clock Mode to mark statistics or halt repetitive execution when the states between two points are greater than a defined number.

**PP -** ***halt_repetitive_execution when_states_x_o***
***greater_than <STATES>***

*when_time_x_o* *time*

Used with the *halt_repetitive_executions* command to specify a period of time to be measured between the mark_x and mark_o samples which, when found, will cause the analyzer to generate the *halt* command for repetitive executions.

**PP -** ***halt_repetitive_execution when_time_x_o greater_than 40 usec***

*when_greater_than* *when*

Used with the *trigger* command (with 16 or more channels) to define a second condition which must be simultaneously true when triggering on a first condition. The second condition requires a pattern that must be true for more than a specified period of time. When the first condition in the trigger specification includes a glitch, the entire specfication will be synchronous and the time will be constrained to be two or more sample periods.

**TS -** ***trigger on entering POD1 = 0 when_greater_than 1 usec_of POD2 = 0***

*wide_sample* *wide*

Used with the *mode_is* command to set the operating mode of the analyzer to sample data on all eight inputs on each probe pod.

**TS, FS -** ***mode_is wide_sample***

*width* *width*

Used with the *define* command to specify the number of bits to be included in a label. The bits start with the one specified and include as many higher numbered bits as specified, limited to the bits available in the hardware.

**FS -** ***define NEWLABEL pod_1_bit 3 width 5***

*with* *with*

Used with the *trigger* command in the Glitch Mode to require simultaneous occurrence of a pattern and one glitch to satisfy the trigger condition.

**TS -** ***trigger on any_glitch on POD1. 0 with POD1 = 0***

*with_data* *with_data*

Used with the *configure* command to store the content of the trace memory along with the trace file containing the Trace, Format, and Post_process Specifications, and the Timing/State Diagram and Trace List formats.

**SD, TS, FS, TD, TL, PP -** ***configuration save_in FILENAME:USERID with_data***

*write_protect* *protect*

Used with the *configure* command to prevent overwriting a measurement configuration file containing Trace, Format, and Post_process Specifications, and Trace List and Timing/State Diagram formats (with or without trace memory data). The write-protect feature does not protect a file from the system commands of **purge** and **copy**.

**SD, TS, FS, TD, TL, PP -** ***configuration save_in FILENAME:USERID write_protect***

*x* *x*

Used with the *mark* command to identify a particular sample within the trace memory for making measurements between samples or for returning to that location easily.

**SD, TD, TL -** ***mark x***

*x1* *x1*

Used with the *magnify* command to select the horizontal magnification that shows the entire memory contents on screen (half contents in Fast Sample Mode). In effect there are four-hundred samples per division. If there are transitions which are too close together to be be displayed at this horizontal resolution, they are shown with a Glitch symbol (broken vertical bar).

**SD, TD -** ***magnify x1***

*x2* *x2*

Used with the *magnify* command to select the horizontal magnification that shows about 50% of the memory contents on screen.

**SD, TD -** ***magnify x2***

*x4* *x4*

Used with the *magnify* command to select the horizontal magnification that shows about 25% of the memory contents on screen.

**SD, TD -** ***magnify x4***

*x10* *x10*

Used with the *magnify* command to select the horizontal magnification that shows about 10% of the memory contents on screen (5% in Fast Sample Mode). In effect, forty samples are shown per division. If there are transitions too close together to be displayed at this horizontal resolution, they will be shown with a Glitch symbol (broken vertical bar).

**SD, TD -** ***magnify x10***

*x20* *x20*

Used with the *magnify* command to select the horizontal magnification that shows about 5% of the memory contents on screen.

**SD, TD -** ***magnify x20***

*x40* *x40*

Used with the *magnify* command to select the horizontal magnification that shows about 2.5% of the memory contents on screen.

**SD, TD -** ***magnify x40***

*x100* *x100*

Used with the *magnify* command to select the horizontal magnification that shows about 1% of the memory contents on screen (0.5% in Fast Sample Mode), at four samples/division.

**SD, TD -** ***magnify x100***

# Appendix B
# ANALYZER SYNTAX DIAGRAMS

## INTRODUCTION

The analyzer syntax diagrams included in this manual are listed below.

**"Define" Syntax Diagram**

**"Threshold" Syntax Diagram**

**"Display" Syntax Diagram For Timing/State Diagram**

**"Magnify" Syntax Diagram**

**"Process_for_data" Syntax Diagram**

**"Halt_repetitive_execution" Syntax Diagram**

**"Copy" Syntax Diagram**

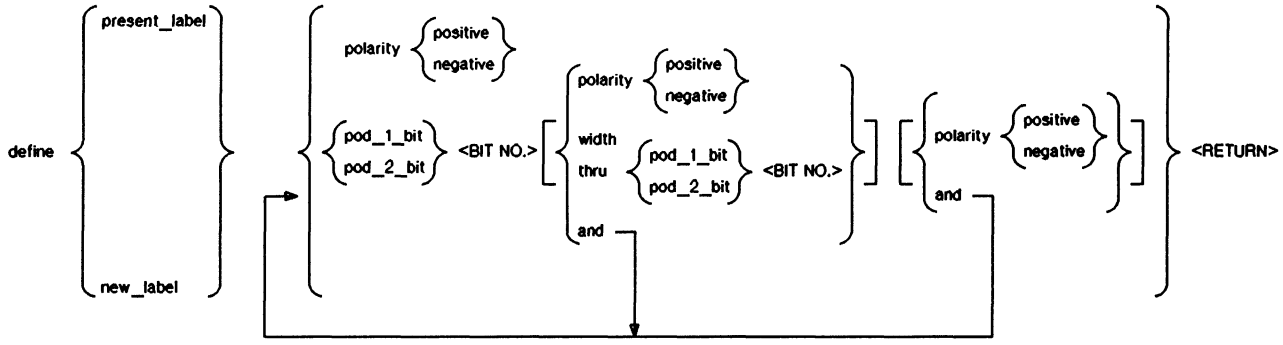
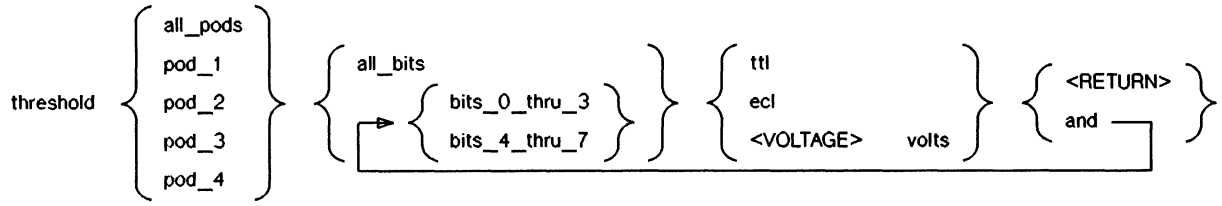
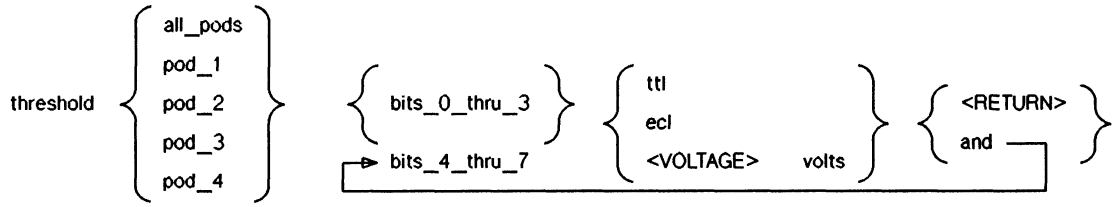



**Figure B-1. "Define" Syntax Diagram**

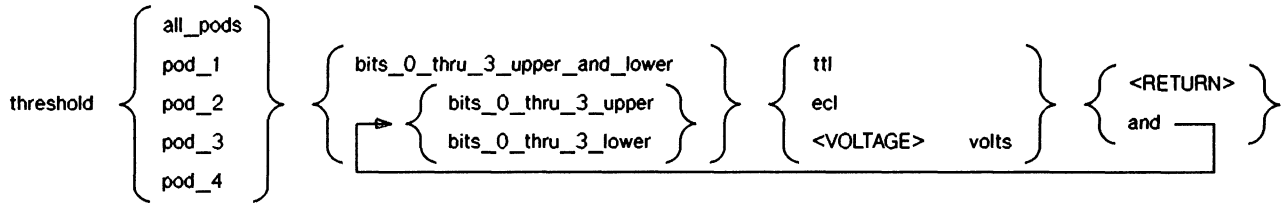
a) WIDE SAMPLE MODE



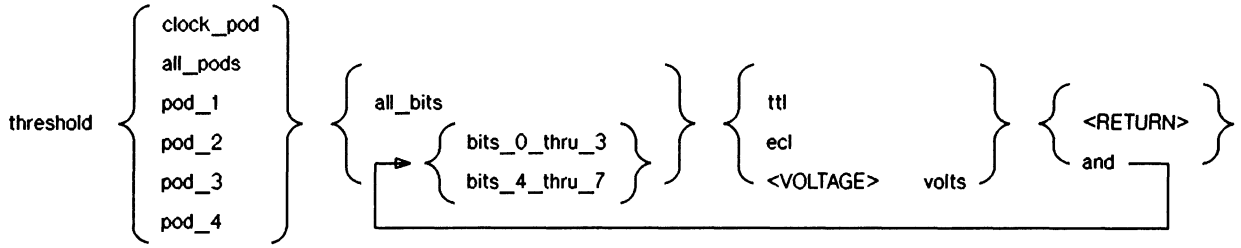

b) GLITCH CAPTURE MODE AND FAST SAMPLE MODE




c) DUAL THRESHOLD MODE




d) EXTERNAL CLOCK MODE




<VOLTAGE> is any number between -10 and +10 volts in 0.1v increments.

**Figure B-2. "Threshold" Syntax Diagram**

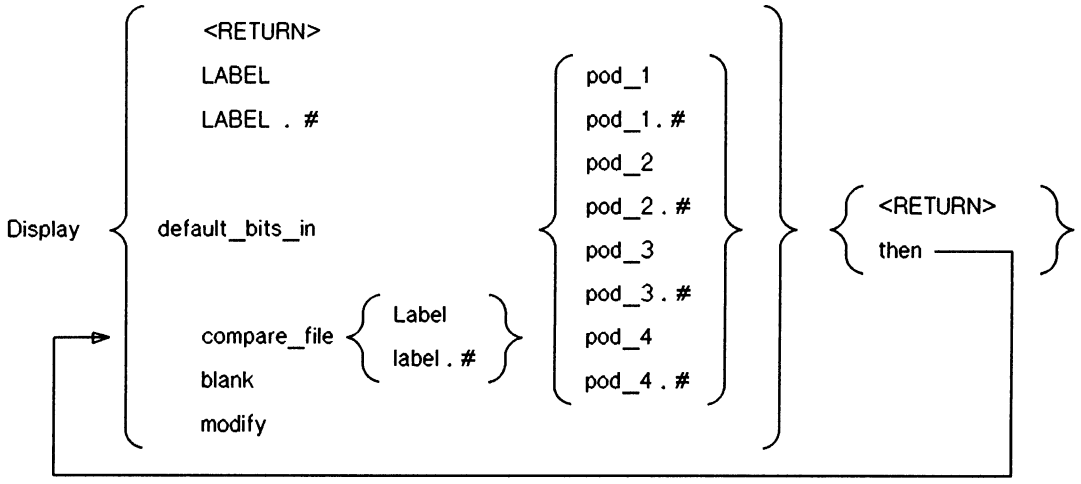



**Figure B-3. "Display" Syntax Diagram For Timing/State Diagram**

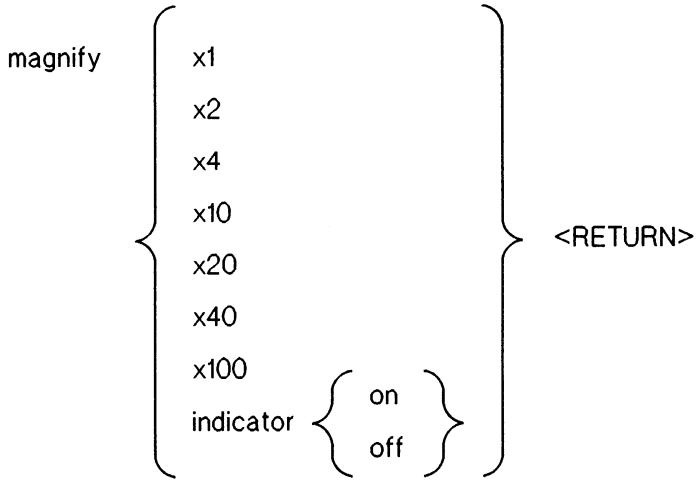
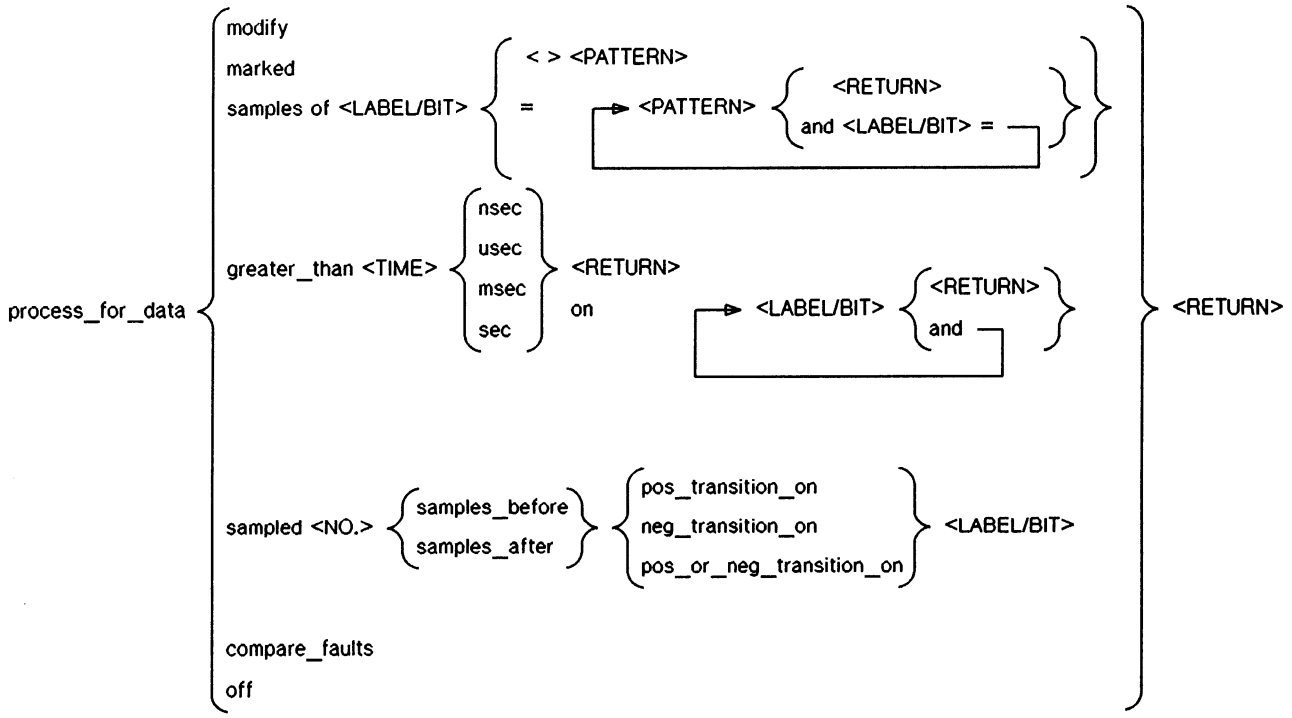



**Figure B-4. "Magnify" Syntax Diagram**




EXTERNAL CLOCK MODE

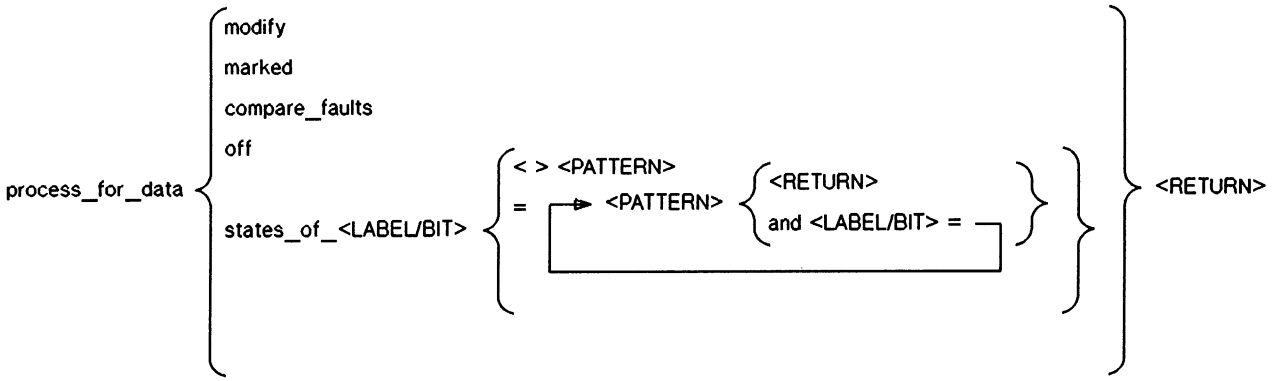
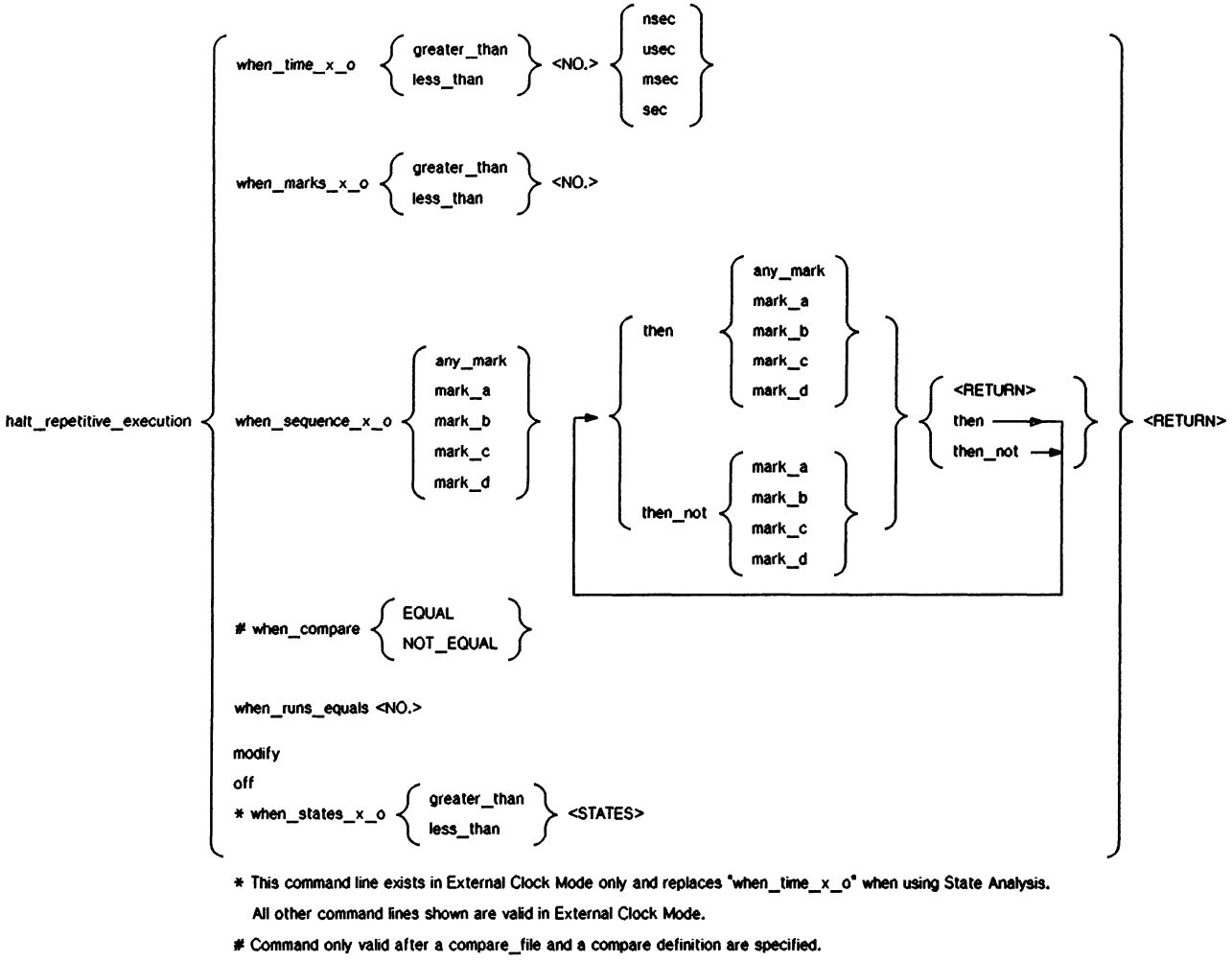


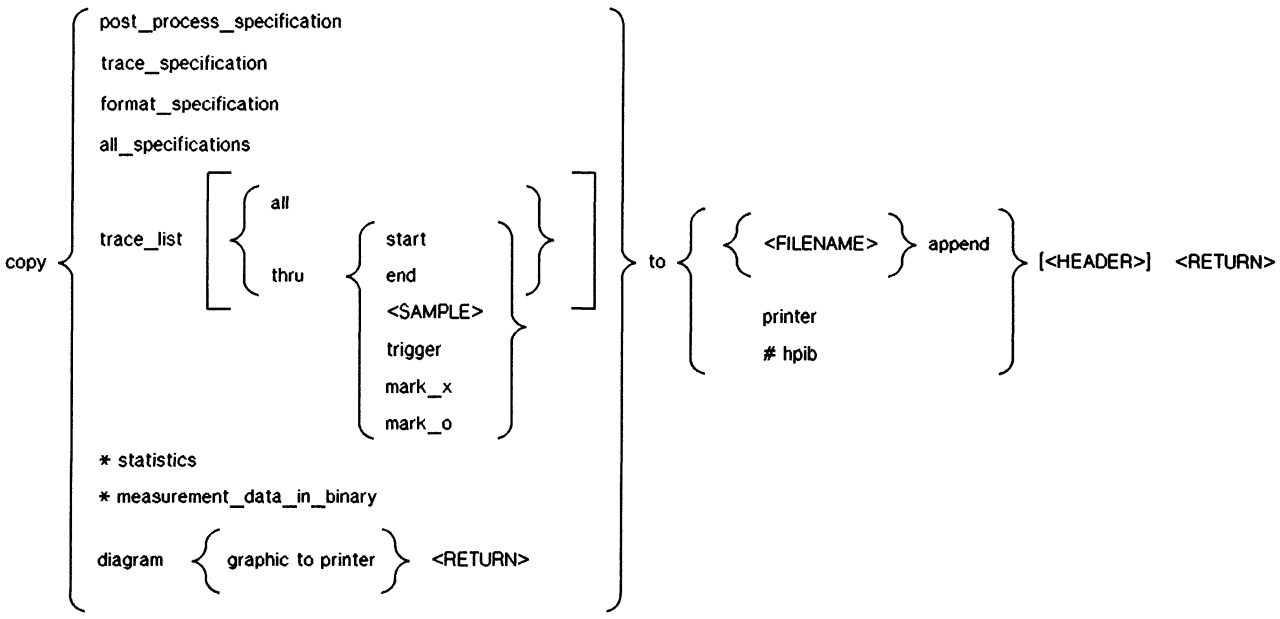

**Figure B-5. "Process_for_data" Syntax Diagram**




* This command line exists in External Clock Mode only and replaces "when_time_x_o" when using State Analysis. All other command lines shown are valid in External Clock Mode.

\# Command only valid after a compare_file and a compare definition are specified.

**Figure B-6. "Halt_repetitive_execution" Syntax Diagram**




* must be in the timing/state diagram or the trace list to copy the statistics and measurement data to a file or printer.

\# hpib used in stand-alone mode only.

**Figure B-7. "Copy" Syntax Diagram**

# Appendix C

## STORING MEASUREMENT DATA IN BINARY

### INTRODUCTION

Storing the measurement data in binary allows you to obtain a compact listing of the measurement data. The following is a description of the contents of a file that contains the measurement data in binary.

When you copy measurement data in binary to a file, a :data type file is created. You can observe the contents of this file by entering ***copy*** **FILENAME:data** ***to display***. You can obtain a listing of the file by copying the FILENAME:data to your printer.

### WHAT IS INCLUDED IN THE DATA FILE?

The data type file is composed of a series of records. Each record can contain 128 words (16 bits/word) maximum. The records (#1 through #N) are described below.

Record #1 - Ascii timing/state descriptor heading record of 29 words.

Record #2 - Ascii comments record of variable length up to a maximum of 120 words.

Record #3 - Ascii measurement_complete date_and_time record of 8 words.

Record #4 through #N - Measurement data in binary format of 128 words (extra words are filled with zeros).

### UNDERSTANDING THE DATA FILE CONTENTS

In all cases, the data displayed in the first display sample of the analyzer corresponds to the first word in record #4. The last display sample occurs somewhere in record #N. Each sample is stored sequentially between these two words (from left to right).

In the **8- and 16-channel** analyzers, each word from record #4 through record #N represents one sample of data. Each word is in the form: BBAA. BB represents the 8 bits of POD2 and AA represents the 8 bits of POD1.

In the **24- and 32-channel** analyzers, two words are used to represent one sample of data. The two words are in the form: DDCC BBAA. DD represents the 8 bits of POD4 (the upper 8 bits of the first word). CC represent the 8 bits of POD3. BB represents the 8 bits of POD2. AA represents the 8 bits of POD1 (the lower 8 bits of the second word).

## ANALYZER MODE DETERMINES DATA FILE FORMAT

The format of each 8 bits of a pod depend on the mode of the analyzer (Wide Sample or External Clock, Fast Sample, Glitch Capture and Dual Threshold Modes).

### Wide Sample or External Clock Mode

In Wide Sample or External Clock Mode, the eight bits represent the values associated with each of the channels (a "1" in a bit position indicates the data sampled is high, a "0" in a bit position indicates the data sampled is low).

B0 = pod_x_bit 0 data
B1 = pod_x_bit 1 data
B2 = pod_x_bit 2 data
B3 = pod_x_bit 3 data
B4 = pod_x_bit 4 data
B5 = pod_x_bit 5 data
B6 = pod_x_bit 6 data
B7 = pod_x_bit 7 data

### Fast Sample Mode

In Fast Sample Mode, the eight bits represent the values associated with each of the channels as shown below. A "1" in a bit position indicates the data sampled is high. A "0" in a bit position indicates the data sampled is low.

B0 = pod_x_bit 0 data
B1 = pod_x_bit 1 data
B2 = pod_x_bit 2 data
B3 = pod_x_bit 3 data
B4 = 0
B5 = 0
B6 = 0
B7 = 0

## Glitch Capture Mode

In Glitch Capture Mode, the eight bits represent the values associated with each of the channels as described below.

A "1" in a data bit position indicates data sampled high.
A "0" in a data bit position indicates data sampled low.
A "1" in a glitch bit position indicates that a glitch occurred.
A "0" in a glitch bit position indicates that no glitch occurred.

B0 = pod_x_bit 0 data
B1 = pod_x_bit 1 data
B2 = pod_x_bit 2 data
B3 = pod_x_bit 3 data
B4 = pod_x_bit 0 glitch
B5 = pod_x_bit 1 glitch
B6 = pod_x_bit 2 glitch
B7 = pod_x_bit 3 glitch

## Dual Threshold Mode

In Dual Threshold Mode, the eight bits represent the values associated with each of the channels as described below.

| Upper Threshold Bit Position | Lower Threshold Bit Position | |
|---|---|---|
| 0 | 0 | = data sampled low |
| 0 | 1 | = data sampled middle |
| 1 | 1 | = data sampled high |
| 1 | 0 | = data sampled error (should not occur) |

B0 = pod_x_bit 0 lower threshold
B1 = pod_x_bit 1 lower threshold
B2 = pod_x_bit 2 lower threshold
B3 = pod_x_bit 3 lower threshold
B4 = pod_x_bit 0 upper threshold
B5 = pod_x_bit 1 upper threshold
B6 = pod_x_bit 2 upper threshold
B7 = pod_x_bit 3 upper threshold

## OBSERVING THE MEASUREMENT DATA IN BINARY

The following is an example of records #1 through #5 in a data file. The words are read by row from left to right.

```
Record #   1    size =  29
2020  3634  3631  3020  5469  6D69  6E67  5F53          64610 Timing_S
7461  7465  2041  6E61  6C79  7A65  7220  2020        tate Analyzer
2020  4D6F  6E2C  2032  3420  5365  7020  3139          Mon, 24 Sep 19
3834  2C20  2038  3A30  3320                        84,  8:03

Record #   2    size =   1
2020

Record #   3    size =   8
2020  2032  3453  6570  3834  2020  383A  3032           24Sep84  8:02


Record #   4    size = 128
0800  FB00  FB00  BF00  4800  FD00  7500  FF00
0000  9F00  BF00  0200  5400  EE00  2F00  8600
0000  BF00  FB00  2100  4800  F500  DD00  2900
4000  EF00  9600  0800  F600  FB00  EF00  8300
2C00  FF00  DB00  4100  FE00  FF00  3F00  8800
7900  EF00  2F00  2D00  EE00  CA00  2600  FB00
E000  9700  DF00  DD00  1A00  3500  2D00  9600
7A00  2B00  2D00  EF00  0800  E700  FF00  FF00
0000  FF00  FF00  FD00  8A00  F500  BF00  BF00
4200  AF00  7F00  C000  C400  BF00  FF00  C700
0000  FB00  FF00  A300  4A00  F700  FF00  2B00
4800  BF00  FF00  0C00  F900  2E00  D200  C100
BC00  B700  D700  0000  FA00  7500  BF00  8600
5000  9F00  BF00  2D00  D200  A200  2E00  FB00
0000  9700  FF00  DD00  2A00  2100  FD00  9600
5100  2F00  BF00  FF00  5C00  EF00  2F00  FF00


Record #   5    size = 128
0000  FF00  FB00  F700  8E00  7500  FD00  BD00
4000  FF00  9700  1A00  4000  FF00  FF00  D300
0000  F700  FF00  B100  4000  7300  FF00  A500
5000  FF00  B700  2000  1100  A200  F300  FB00
0000  8F00  CB00  1500  5800  2300  7700  8E00
5000  3F00  AF00  FF00  4000  6B00  2900  FF00
```

## OBSERVING THE TRACE LIST

This is the beginning of the trace list that corresponds to the measurement data stored. Note that the first sample contains the first 2 words in the data file.

```
64610 Timing Analyzer          Mon, 24 Sep 1984,  8:17

      Trace List              Timing_State 4, 16 ch, 200/400MHz/125MHz
   WIDE_SAMPLE MODE              5 nsec/sample       Time x_o

Label:      POD2       POD1      time count
Base:       bin        bin          abs

trigger __11101111  00000000   0.0 nsec    __
+0001     10000011  00000000   5.0 nsec
+0002     00101100  00000000   10.0 nsec
+0003     11111111  00000000   15.0 nsec
+0004     11011011  00000000   20.0 nsec
+0005     01000001  00000000   25.0 nsec
+0006     11111110  00000000   30.0 nsec
+0007     11111111  00000000   35.0 nsec
+0008     00111111  00000000   40.0 nsec
+0009     10001000  00000000   45.0 nsec
+0010     01111001  00000000   50.0 nsec
+0011     11101111  00000000   55.0 nsec
+0012     00101111  00000000   60.0 nsec
+0013     00101101  00000000   65.0 nsec
+0014     11101110  00000000   70.0 nsec
+0015     11001010  00000000   75.0 nsec
+0016     00100110  00000000   80.0 nsec
+0017     11111011  00000000   85.0 nsec
+0018     11100000  00000000   90.0 nsec
+0019     10010111  00000000   95.0 nsec
+0020     11011111  00000000   100.0 nsec
+0021     11011101  00000000   105.0 nsec
+0022     00011010  00000000   110.0 nsec
+0023     00110101  00000000   115.0 nsec
+0024     00101101  00000000   120.0 nsec
+0025     10010110  00000000   125.0 nsec
+0026     01111010  00000000   130.0 nsec
+0027     00101011  00000000   135.0 nsec
+0028     00101101  00000000   140.0 nsec
+0029     11101111  00000000   145.0 nsec
+0030     00001000  00000000   150.0 nsec
```

# Appendix D

# PERFORMING ACCURATE TIME INTERVAL MEASUREMENTS

## INTRODUCTION

This appendix provides information for making accurate time interval measurements in the timing analyzer.

Accurate time intervals depend on the time interval resolution of the timing analyzer. If you have good time interval resolution, using statistics can improve the accuracy of a measured time interval by the amount described in the following equations. Accurate statistical measurements can only be made if the input intervals are a uniform distribution of the interval to be measured. The following material describes each of the aspects of making accurate time interval measurements.

## TIME INTERVAL RESOLUTION

Time interval measurement resolution is one of the most important specifications of a timing analyzer. The resolution depends on the factors outlined below:

1. Sample Period
2. Interchannel Skew
3. Memory Depth

When measuring a time interval, the **resolution** is defined as:

**resolution = plus or minus (sample period + skew)**

**Skew** is the difference in delays of the probe channels, including delay differences from one channel to another, and delay differences in recognizing negative and positive transitions.

The high sample rate (400 MHz) coupled with low skew (plus or minus 1.5 nS for both opposite and same direction transitions) give the timing analyzer very good resolution.

**Skew** is a function of several input variables as outlined below:

1. Input signal slew rate in volts/nS (low slew rate increases skew).
2. Signal overdrive above the threshold as a % of skew (low overdrive increases skew).
3. Threshold value selected (high threshold settings increase skew).

The **skew specification** of 1.5 nS for all probes within one pod is measured according to the following conditions:

1. A 0.25-volt per nS slew rate.
2. A 0.6-volt amplitude signal with equal swings on either side of threshold.
3. A minus 1.3-volt threshold.

Memory depth is also important in time interval resolution. The memory depth sets the maximum time interval that can be measured with any sample period. In the 400 MHz (Fast Sample) mode of operation, a time interval of 20.4 uS can be measured with full sample accuracy, for example:

**(8140 samples of memory depth) (2.5 nS per sample) = 20.4 uS**

To measure longer time intervals, the sample period must be increased.

## IMPROVING THE ACCURACY OF TIME INTERVAL MEASUREMENTS

You can improve the accuracy of an interval measurement by making the measurement with a series of repetitive executions. When a single execution is made, the measurement accuracy is equal to +/- the sample period + skew. When measuring a stable interval using a series of repetitive executions, the accuracy of the measurement improves by the following formula:

**accuracy = +/- (sample period) / sqrt(n) + skew**

**sqrt(n)** = the square root of the number of executions included in the measurement.
**sample period** = the sample period specified in the timing analyzer trace specification.
**skew** = the delay differences between input channels.

### NOTE

The time interval being measured must not be synchronous to the sampling clock of the timing analyzer. This is typically not a problem in the timing analyzer unless the sample rate is extremely slow. The timing analyzer halts its interval sample clock between each measurement, therefore the probability is low that the time interval being measured is synchronous with the timing interval sample clock.

With the formula above, an improvement of x10 is obtained in the accuracy of your measurement when using 100 repetitive executions to obtain the measurement.

## IMPROVING THE ACCURACY OF MEAN VALUE MEASUREMENTS

The accuracy of the displayed mean value of a single interval depends on the number of executions in the series used to determine the mean value.

Assume the timing analyzer is measuring a stable, repetitive time interval approximately 100 uS long. Using a 20 MHz sample rate, you capture 203 uS of timing data due to the following equation:

**4060 * 50 nS = 203 uS**

**4060** = the depth of the timing analyzer memory
**50 nS** = the selected sample period

A single measurement will have the following accuracy:

**+/- (50 nS + 1.5 nS) = +/- 51.5 nS**

**1.5 nS** = the skew specification for a single pod
**50 nS** = the sample period

By making 100 measurements in a repetitive series, the accuracy of the mean value displayed will be improved by a factor of the square root of the number of traces included in the series, as indicated by the following formula:

**+/- [(50 nS/sqrt(100) + 1.5 nS] = +/- 6.5 nS**

## ACCURACY OF STANDARD DEVIATION MEASUREMENTS

An interval that does not vary can still be shown to have a large standard deviation due to the sampling process. The error in the displayed standard deviation depends on the size of two elements: (1) the portion of the interval that exceeds the multiple of the sample periods, and (2) the portion of the interval that includes complete sample periods.

**Example 1:** Assume the timing analyzer is measuring a time interval of exactly 12.5 nS. It makes 10 executions using a 5 nS sample period. Five of the executions show the interval to last 15 nS (3 sample clocks), and five of the executions show the interval to last 10 nS (2 sample clocks). Even though the input signal has a true standard deviation of 0.0 nS, the timing analyzer will calculate the standard deviation of this signal to be 2.64 nS and display this standard deviation on screen. When the sampled standard deviation is less than one sample period, its value is mainly determined by the sampling process.

**Example 2:** Assume the timing analyzer is measuring a time interval that varies from 5 to 8 uS. The timing analyzer is operating with a 5 nS sample period. After a series of repetitive measurements, the timing analyzer shows a standard deviation of 1.5 uS for the interval being measured. This dispersion is determined by variations in the time interval itself, and not the sampling process. In this case, the standard deviation is much larger than one sample period.

## STATISTICAL ERRORS CAUSED BY SAMPLING PROCESS

The timing analyzer calculates statistics on the sampled data in its memory, not on all of the data generated by the system under test. The data in memory may misrepresent the actual data. Misleading data can be captured when you trigger your trace on some occurrence that causes the timing analyzer to capture samples at misleading points in the data flow of the system under test.

Use of the "trigger on anything" specification may not overcome all measurement bias problems. Consider the case where the timing analyzer is measuring an interval of time between positive edges occurring on a probe line. Suppose there are two intervals on that line, and they are occurring alternately (one is 10 uS long and the other is 20 uS long). Interval measurements are made by marking "x" on the first positive edge of the selected label and marking "o" on the next positive edge after "x". The random beginning of a new trace will probably occur twice as often during the 20 uS interval as during the 10 uS interval. Because of this, the timing analyzer will appear to be finding twice as many 10 uS intervals as 20 uS intervals, but in the system under test there are equal numbers of 10 uS and 20 uS intervals.

One possible approach to solving the problem of misleading data in the above example is to find another line with a uniform square wave operating at twice the frequency of the combined intervals. Such a square wave will have as many positive edges preceding 10 uS intervals as 20 uS intervals. By triggering the interval measurements on positive edges in that square wave, and marking "x" and "o" on the first interval after each trigger, the timing analyzer will measure as many 20 uS intervals as 10 uS intervals.

# Appendix E
## ANALYZER SPECIFICATIONS

### SPECIFICATIONS

Specifications for the High Speed Timing/State Analyzer are listed below. These specifications are the performance standards or limits by which the instrument is tested. The specifications listed here are provided for the Model 64601B Control Board, the Model 64602A Acquisition Board, the Model 64604A Probe, and the Model 64605A Clock Probe.

**RESOLUTION:**

Overall resolution: +/-(sample period + skew)
Total skew from probe tip:
Within pod: +/- 1.5 nS.
Pod to pod: +/- 3.0 nS.
These specifications are true for these conditions:
Input signal: VH = -1.0V, VL = -1.6V, VTH = -1.3 V.
Input slew rate: greater than 0.25 V/nS (without delay module).
Sample rate accuracy: typically +/- 0.002% - adjustable from 2 Hz to 400 MHz.

**PROBE CHARACTERISTICS (Models 64604A and 64605A):**

Input impedance: 100K ohms +/- 2%, shunted by less than 6 pF.
Drive requirements:
Minimum input amplitude - 600 mV P/P.
Minimum input overdrive - 200 mV or 25% of input amplitude, whichever is greater.
Minimum input pulse width - 3.0 nS at threshold.
Dynamic range: +/- 10 V.
Maximum input: +/- 40 V.
Threshold accuracy: +/- 50 mV or +/- 2%, whichever is greater.
Hysteresis: Typically 50 mV.

**MEMORY DEPTH:**

Wide Sample, Glitch, Dual Threshold and External Clock Modes: 4k
Fast Sample Mode: 8k

**GLITCH MODE:**

Maximum sample rate - 100 MHz.
Minimum glitch width - 3.0 nS at threshold without data delay module.
Maximum glitch width - sample period less 4.0 nS.

**TRIGGERING:**

Time duration accuracy: +/- (20% + 2 nS).
Time duration for restart: For accurate restarts, pattern must go false for at least 1.5 times the selected time duration.
Minimum width for "narrower-than" trigger - 5 nS typical.
Minimum width for transition trigger - 5 nS typical.
Displayed position accuracy:
+/- 2 samples in Wide Sample, Dual Threshold and Glitch Modes.
+/- 4 samples in Fast Sample Mode.

**TRIGGERING (Cont'd):**

Delay from input to external BNC drive - typically 55 nS.
Delay from input to internal IMB drive - typically 50 nS.
Dead time for restart measurement to reset:
Typically 50 nS plus the time required to fill the memory with the selected amount of pre-trigger information.

**EXTERNAL CLOCK MODE:**

Clock Frequency: 0 to 125 MHz maximum
Minimum Width: 3 nS at threshold
Setup Time: 4 nS maximum data to clock, 4 nS maximum qualify to clock
Hold Time: 0.5 nS maximum data to clock, 0.5 nS maximum qualify to clock
Clocking Capabilities: rising edge, falling edge with or without HI or LO level qualify.
Triggering: The trigger is synchronous, and therefore the trigger condition must meet the setup and hold time requirements.
These specifications are true for these conditions:
Input signal: VH = -1.0V, VL = -1.6V, VTH = -1.3 V.
Input slew rate: greater than 0.25 V/nS.

**BNC DRIVE:**

Output signal swing in transition trigger mode:
Amplitude - 2.0 V typical with 50 ohm load.
Width at 50% - 10 nS typical.
Output signal swing in "width greater-than" trigger mode:
Amplitude - 2.5 V typical.
Width - Input trigger width minus the selected duration.
Output signal swing in width less-than trigger mode:
Amplitude - same as in transition trigger mode.
Width - same as in transition trigger mode.
Position - occurs when trigger pattern disappears, before the selected duration times out.

**IMB FUNCTIONS:**

Master Enable ..... Drive, Receive (Execute/Halt only).
Trigger Enable ..... Drive, Receive.
Trigger ..... Drive, Receive.
Delay Clock .....Receive only.
Storage Enable ..... Not used.

**POWER SUPPLY REQUIREMENTS:**

| | +5V | +12V | +17V | -3.25V | -5.20V | -12V |
|---|---|---|---|---|---|---|
| CONTROL | 1.60A | - | - | 1.50A | 2.10A | .02A |
| ACQUISITION | 3.60A | - | - | 2.50A | 0.80A | - |
| PROBE | 0.04A | - | - | - | 0.24A | - |
| CLOCK PROBE | 0.04A | - | - | - | 0.08A | - |
| totals | 5.28A | - | - | 4.00A | 3.22A | .02A |
| (timing only) | 5.24A | - | - | 4.00A | 3.14A | .02A |

# Appendix F

## THEORY OF OPERATION

### INTRODUCTION

This is a brief overview of the theory of operation of the High Speed Timing/State Analyzer. The theory is presented in two parts: hardware theory and software theory.

### HARDWARE THEORY

The hardware of the timing/state analyzer is composed of standard and custom ECL circuits working in unison to capture data at extremely fast rates. The design of these circuits is such that channel to channel uniformity is maximizied. Channel uniformity implies that the channel to channel skew is as low as possible so that all channels are sampled at nearly the same instant in time. This is accomplished by making the data paths of identical lengths and using custom integrated circuits to bundle all channels and associated circuitry onto one substrate. In addition, the data channels and sampling clock are driven differentially to minimize rising and falling edge effects. The internal sampling clock is also crystal controlled to ensure that durations between samples are identical. The basic components of the timing/state analyzer include the data probes which provide input of the data, the data acquisition boards which store the captured data, the control board which controls the measurement and the clock probe which provides external input of the clock for state analysis (see figure E-1).

Starting with the data probes, the input data is passed through a high impedance probe to a comparator circuit which determines if the signal is above or below a reference signal. The comparator circuit is built using a custom integrated circuit so that all eight comparators can be located on one substrate. In Wide Sample Mode all the channels are compared against a single threshold and data is sampled at a user specified rate from 200 MHz down to 2 Hz. In Dual Threshold Mode the upper four channels are disconnected and the lower four channels are compared against two different thresholds. The sampled result of this comparison produces a two bit data pattern for each data sample for each channel with the following format:

below upper threshold, below lower threshold = 00 or low
below upper threshold, above lower threshold = 01 or middle
above upper threshold, above lower threshold = 11 or high

In Fast Sample Mode the upper four channels are disconnected and the lower four channels are sampled again but this time the 200 MHz sampling clock is delayed by 2.5 nanoseconds. This produces the effect of sampling the same data twice at 200 MHz but with a 2.5 nanosecond shift between each of the sampling points yielding an effective sampling rate of 400 MHz. In Glitch Capture Mode the upper four channels are disconnected and the lower four channels are examined between samples by a multiple transition or glitch detection circuit. The result of this glitch detection is then stored for each sample for each channel. In External Clock Mode all the channels are compared against a single threshold and data is sampled by the user input clock on the appropriate edge. The qualify function works by inhibiting the clock generation if the qualify is not in the level specified.

The timing/state analyzer samples data both asynchronously (timing) and synchronously (state). In asynchronous sampling the data is sampled by the internal sample clock which is set at a rate specified by the user. In synchronous sampling the external clock probe supplies the clock for the analyzer to sample the data. The data delay modules must be inserted between the data probes and the data cables to assure that the external sample clock will be sampling the correct data from the data probes (If the data delay modules are left installed while using timing analysis, the only affect will be an increase in the minimum glitch width, which can be detected in Glitch Capture Mode. All other timing analysis is not affected).

Sampling of data occurs on another custom chip called the glitch chip which is included on each data acquisition board. The glitch chip provides uniform sampling and glitch detection. It also determines if pattern matches occur and produces a pattern match signal which is sent via a timing bus to the control board. In the state analyzer, the pattern match signal is sampled with the sampling clock or external clock. In the timing analyzer, the pattern match signal is not sampled with the internal sampling clock (except in Glitch Trigger Mode). The pattern match function operates separately from the sampling clock to enable trigger recognition on narrow phenomena which would not likely be sampled, especially when using a slow sampling clock rate. The trigger position is then marked on the screen even though the trigger event may not be sampled and thus not displayed.

In the timing analyzer, trigger events must be present at the probe inputs for 6 nS (typical) to be recognized, except for glitch triggers in Glitch Capture Mode. In Glitch Capture Mode, the minimum width for a trigger (3 nS) is set by the glitch capture circuitry. Once a data pattern has been sampled by the glitch chip, it is stored into data acquisition or trace memory via a data rate reducer (serial to parallel converter) custom integrated circuit. The trace memory is circular, meaning that new data is always rewriting the oldest data. A halt signal from the control board halts the storing of data after the trigger condition has been found and the appropriate number of samples stored. The data is then read out of the data acquisition board (from trace memory) to build the diagram and trace list.

The control board receives up to four pattern match signals (one from each data acquisition board) and determines if the trigger conditions are met (duration trigger is correct, logic combination of patterns is correct, trigger enable satisfied, trigger sequence satisfied) and produces a trigger pulse. This pulse is the same pulse which can be asserted to the bnc port 4. The trigger pulse is then sent to a delay circuit which adds the specified delay. After the specified delay is counted out, the trigger occurs and the trigger location in trace memory is marked. This trigger can also drive the IMB trigger or IMB trigger_enable. In the case of driving the IMB trigger, the trigger location is modified to be the trace sample in time when the trigger was driven to the IMB bus. In the case of receiving the IMB trigger, the trigger location is the trace sample in time when the trigger was received from the IMB bus.

The effect of modifing the trigger location to reflect the point in time when the trigger was seen on the IMB bus allows multiple timing/state modules to be closely coupled in time in their display of trigger events (typically less than 2 samples apart). This means that a 32 channel timing/state analyzer can be closely coupled with another 32 channel timing/state analyzer to yield trace_lists which are closely correlated. The trigger signal is then passed to the halt counter. The halt counter counts the number of samples after the trigger condition to achieve a full trace before it sets the halt signal true. A user specified halt will set this halt signal true if the measurement has not completed. The trigger location can not be found whenever a measurement is halted.

The clock probe compares the input clock signal against the clock threshold and produces a clock signal. This clock signal then clocks in the data at the data probe tips on either the rising or falling clock edge as specified. The clock is inhibited if the clock qualifier is specified and the qualified signal is false. If timing analysis is requested, an internal crystal is the source of the clock signal. The frequency of the sampling clock is determined by dividing the crystal frequency to obtain the desired sampling frequency.

In the state analyzer, trigger events and data patterns must meet the minimum setup and hold times specified (4.0 nS setup to external clock and 0.5 nS hold after external clock edge).

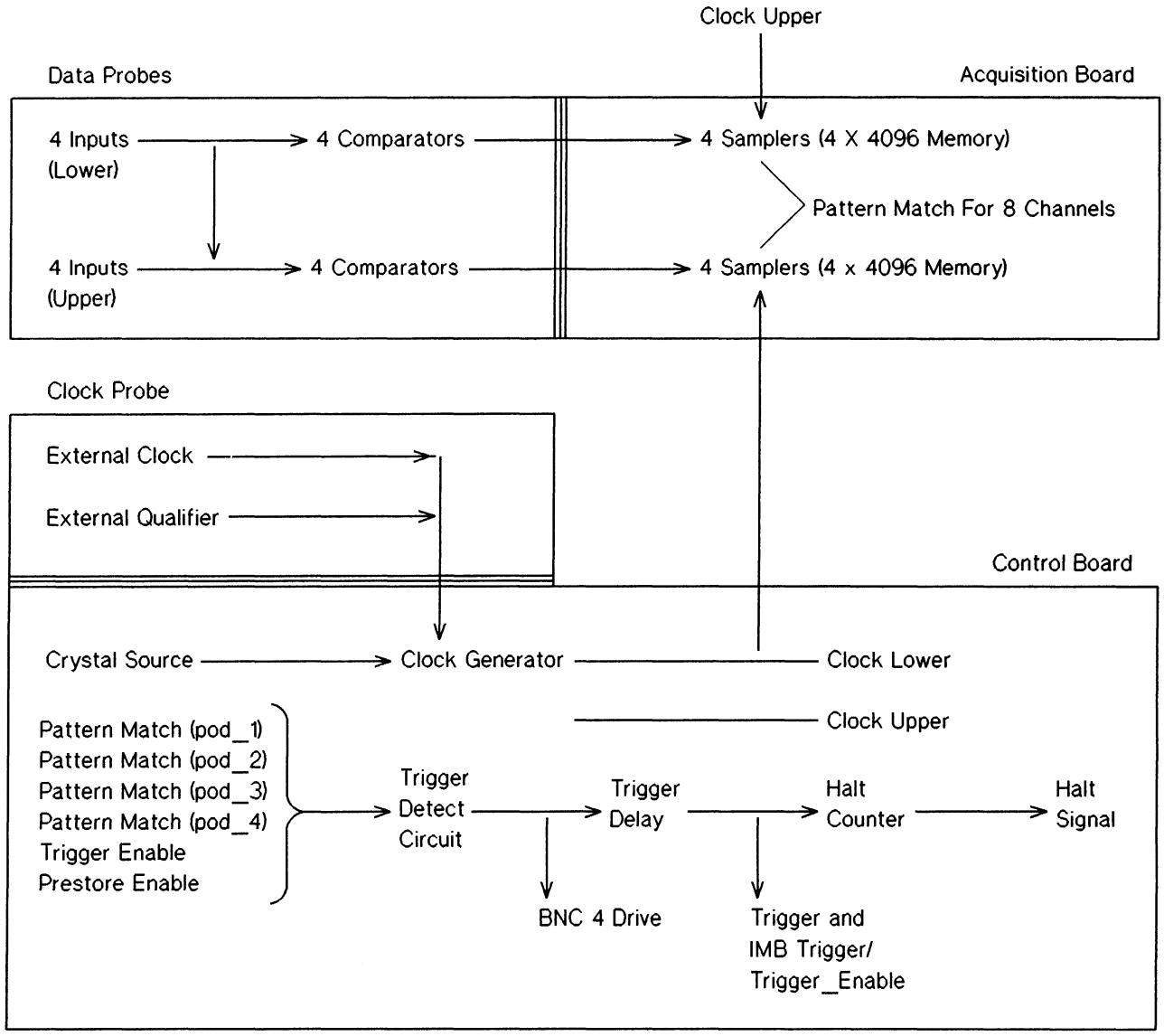



**Figure F-1. Hardware Block Diagram Of The Timing/State Analyzer**

## SOFTWARE THEORY

The software that controls the timing/state analyzer can be thought of as two modules. The first is responsible for running, determining status, and halting the timing analyzer. This segment of code is resident at all times after the measurement_system or timing_state key is pressed. This allows the timing/state analyzer to be executed by another module when it is connected through the intermodule bus and the code of the other module controls the HP64000 development station.

The second module comprises the timing/state analyzer. This module is the resident software which is entered when the user selects the timing/state analyzer. This module is responsible for the interaction with the user, through keyboard and display for input of measurement configurations, for execution of the measurement, for processing the resultant data and final display of the output.

The operation of the timing/state analyzer can be described as a translation of a user imagined measurement to a hard copy output (see figure E-2). The first part of this translation involves the user entering commands which configure the analyzer to make the desired measurement. These commands translate the user desired measurement into a software structure which defines the measurement. This software structure can be readily input via a command file or loaded via a configuration file. Translation of this software structure to the actual hardware setup is accomplished by the loader which is invoked when either the ***execute*** or ***end*** key is pressed. The loader may be skipped if the hardware setup matches the software configuration. If the loader catches an error in the translation of the software structure, it will flag the error and abort the execution to either the trace_specification or the post_process specification, as it determines.

After a successful translation of the software structure to the appropriate hardware setup, the hardware is told to run and capture a trace. When the hardware completes, the translated measurement now resides in the form of raw data in the trace memory of the analyzer. Unloading of this data can not occur until all IMB modules which are executing together are complete. In the normal situation of only one module executing at a time, this means data can be unloaded instantly. Even in the case of multiple modules executing together this amount of time is usually negligible. In the event that the user has not specified any post_processing commands (***compare, mark,*** or ***process***) the data is then translated to the appropriate display format and the user is presented with the results of the envisioned measurement.

When the user specifies any one of the three basic post_processing commands (***compare, mark,*** or ***process***) other translations of the data occur before the user is presented with the data. The first post_processing command to be executed is the compare operation (if it is specified). Compare unloads the trace memory data into a reserved section of space on the memory expander board and compares it to the previously unloaded compare_file data which resides in a second section on the memory expander. Note that the compare_file data is unloaded in such a format that if compare_file label POD1 is to be compared with the current label POD2, then compare_file label POD1 is unloaded into bits which correspond to POD2 of the current label. When the compare occurs it will attempt to match a 32 bit word of current data to a 32 bit word of compare_file data. This is why the command ***compare POD1 to_compare_file POD1 and POD2 to_compare_file POD1*** can work (because the compare labels are mapped to the current labels). Commands like ***compare POD1 to_compare_file POD1 and POD1 to_compare_file POD2*** will not work (because the analyzer can not do a multiple mapping of compare_labels POD1 and POD2 to current label POD1). The compare operation results in an array of 4060-8140 bits (each bit corresponds to one sample) with each non compare resulting in a bit being set in this array for the given sample. The array is then searched for consecutive faults allowed and then modified accordingly so that the end result is an array of bits which indicate the compare_faults. The halt_repetitive_execution is checked to determine whether or not the halt_condition for compare_faults is met.

The marking operation is then executed if marks are defined, if some type of statistics is desired, or if halt repetitive execution is defined for an interval or a number of marks. The marking module uses the data previously unloaded by the compare operation (or earlier post_process operation) or unloads the data from the trace memory into a format suitable for searching for events in the trace memory. Once the data is unloaded, the marking operation begins by first determining if each defined mark is a valid mark definition for the mode selected. If the mark is valid, a one pass search is made through the memory for each of the appropriate mark definitions. This one pass searches for all of the six marks at the same time.

As mark x and mark o are found, they either inhibit or enable the other marks. This is how "**before**" and "**after**" conditions are found. As marks a, b, c and d are found, a buffer is filled with the type of the mark and sample number of the mark location. This buffer is only 511 words long, hence there are only a maximum of 511 events that can be marked. When the data is displayed, this buffer is searched and the appropriate sample numbers are appended with the appropriate mark tags. After the marking of the measurement data occurs, the mark statistics are calculated. If these statistics are to be logged to a file, they are recorded to the file at this time. Note that statistics can only be logged to a file if "***indicate time/states/marks max_min/mean_stdv***" is specified. Finally, the halt repetitive execution condition is checked and a halt_flag is set if the repetitive execution is to be halted.

The third post_processing operation is the process_for_data operation which occurs only in the trace_list. Here it may be desirable to reduce the amount of data that the user must observe by using a single command. Process_for_data works similiar to both the compare and mark commands in that it first unloads the trace memory if needed. The operation of processing for data revolves around determining if a sample is a valid display sample. A 4060-8140 bit array specifies if a sample is valid for display or not. If no process_for_data command is entered, each bit of this array is set true which makes all samples valid. If any of these bits are set false, the corresponding samples will be skipped (or not displayed). The process_for_data module is provided for making these bits false. Each sample which does not meet the specified process_for_data command forces its corresponding bit false. The trace_list display routine then consults this valid_sample array in order to determine how to build the trace_list. Since the process_for_data operation is only specific to the trace_list, the process_for_data command will only be executed if the user is viewing the trace_list.

The two data display outputs include a diagram and a trace_list. The presentation of the diagram data involves building a waveform screen three pages in width. This allows the user to access data within these three pages without having to wait for an unloading of the data for each sample. When a page limit is reached and a sample beyond the current page needs to be displayed, the 3-page buffer is rebuilt with the current screen as the center page. This rebuild time can be noticed as a slight pause as the timing/state diagram is rolled. In the same way, the trace_list is built about a seven page screen and a slight pause may be noticed as this seven page screen is rebuilt.

In summary, the translation process from an imagined measurement to a final measurement results in the task of the timing/state analyzer both through hardware and software. Hopefully with a little understanding of the information presented here, you will be able to more readily make your desired measurements.

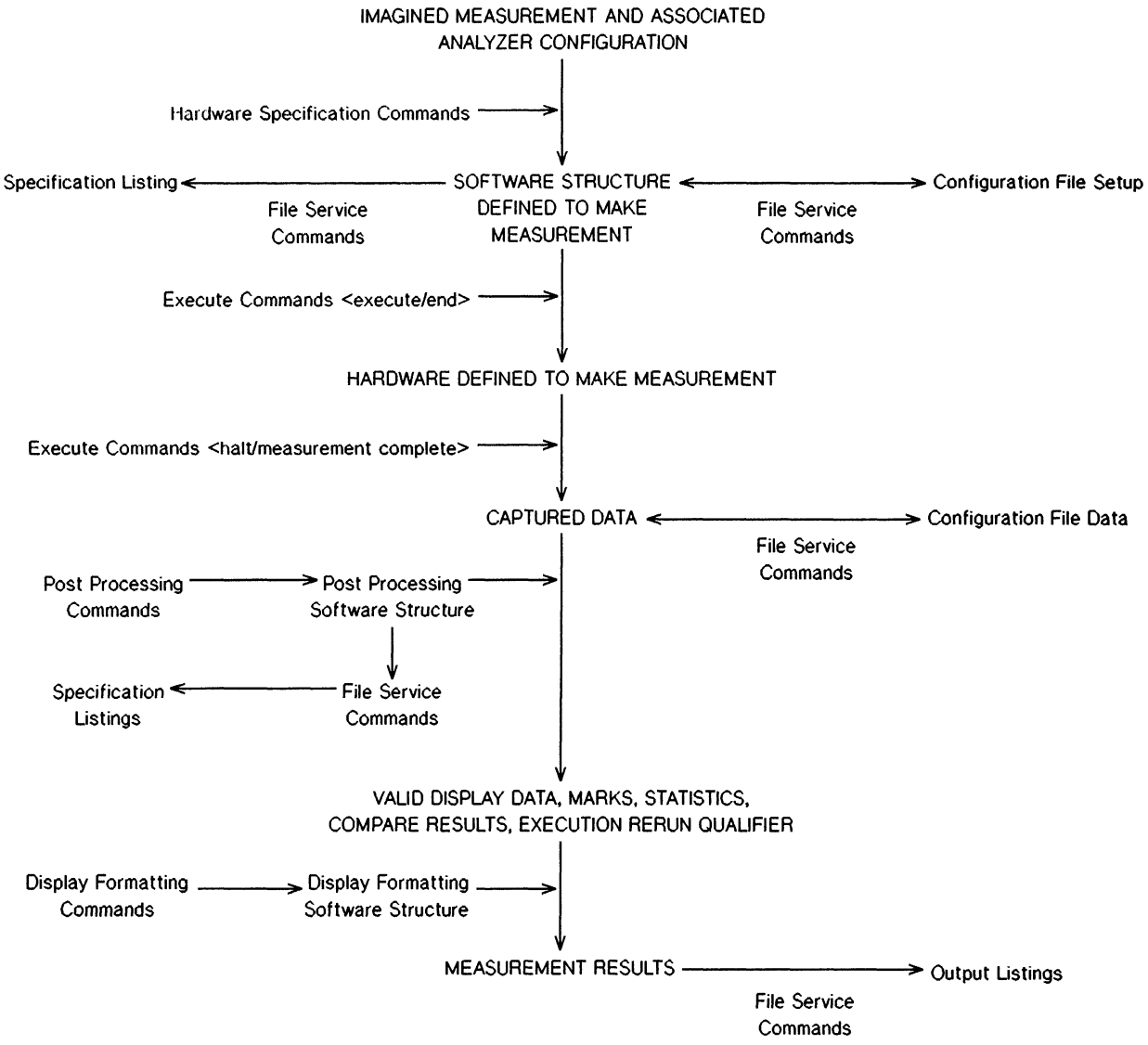



**Figure F-2. Measurement Translation Block Diagram**

**Hardware specification commands:**
Mode, Sample, Trigger, Assert, Default,
Threshold, Clock, Define, Delete, Modify, Rename, Activity_test

**Display formatting commands:**
Display, Diagram, Magnify, Indicate, Cursor, <ROLL>

**Execution commands:**
Execute, Halt, End

**Post processing commands:**
Find, Mark, Compare, Process_for_data, Halt_repetitive_execution, Statistics

**File service commands:**
Configure, Copy

# Appendix G

## ANALYZER ERROR AND STATUS MESSAGES

The following messages are displayed on screen to provide an indication of operating status. Some of these also advise of improper operating conditions or invalid entry on the command line.

**"and" is not possible, all bits are already specified** - Displayed when you try to enter a trigger specification which uses the same bit or bits in two or more labels. Your specification must specify only one entry for each bit.

**BNC port 4 already driven by another module** - Displayed when you try to drive BNC port 4 from the analyzer and it is already being driven by another module.

**Checksum error, file is corrupt** - Displayed when a file is loaded from memory and the checksum on the file does not agree with the checksum calculated by the analyzer during the load operation.

**Command is not valid during an execution** - Displayed when you try to enter a specification command which is invalid during an execution.

**Compare file is invalid, measurement was halted** - Displayed when you try to define a halted measurement as a compare file.

**Compare file labels fill the label table** - Displayed when the configuration named as the compare file has enough labels assigned in it to fill the capacity of the analyzer (32 labels, maximum) when combined with the labels in the current configuration.

**Compare file spec does not agree with hardware** - Displayed when the data in the trace memory was obtained by using a different sample rate, trigger position, or mode than was used to obtain the data in the compare file.

**Data is not present in file** - Displayed when you try to specify a file to be used as a compare_file that was not stored with data.

**Data is not present in hardware** - Displayed when you command the analyzer to save its configuration with_data, and there is no data in the trace memory.

**Data label <YOUR LABEL> bits are already specified** - Displayed when your trigger specification includes the same bit or bits in two or more labels in your trigger specification.

**Data label <YOUR LABEL> is not a valid entry** - Displayed when your trigger specification includes a label which is not valid for the current mode which is set up.

**Data label <YOUR LABEL> specified bit does not exist** - Displayed when your trigger specification includes a label-bit that is not defined in the Format Specification.

**Disc 0 is down** - Displayed when you try to write a file to or load a file from disc drive 0, and the disc drive is not operating.

**Disc 0 is full -** Displayed when you try to save a display or configuration on disc 0 when it has no storage space left.

**Disc 0 is full, writing to disc 1 -** Displayed when you try to save a display or configuration and the analyzer finds disc 0 full but space available on disc 1. It automatically switches the save function to disc 1 and processes your save command.

**Disc 1 is down -** Displayed when you try to write a file to or load a file from disc drive 1, and the disc drive is not operating.

**Disc 1 is full -** Displayed when you try to save a display or configuration on disc 1 when it has no storage space left.

**Duration is greater than trace memory -** Displayed when the entered duration is greater than the duration of the entire trace memory.

**Duration is less than sample period -** Displayed when entered duration is less than the sample period. The analyzer cannot find durations which are less than the sample period.

**Expression too long, shorten mark and trigger expressions -** Displayed when the combination of all the trigger definitions and mark definitions are too long. Shorten or remove unused definitions.

**File exists, but is not a timing analysis file -** Displayed when you try to save a configuration or load a configuration from a filename that already is used for storage of a configuration of some other instrument, such as a state/software analyzer.

**File is write protected -** Displayed when you try to save a configuration under a file name that already contains a write-protected configuration.

**File version <FILE NO.> is incompatible with software version 2310 -** Displayed when you attempt to use a timing analysis configuration file with a later date (future version) for comparison with the current version of software.

**Filling trace memory -** Displayed when the analyzer is filling its trace memory in order to satisfy the prestore requirements. Triggers are not acknowledged during this time.

**"followed_by" is not possible, all pods are used -** Displayed when your trigger specification requires a condition to be found followed by another condition, and both conditions include the same probe bit or bits. The analyzer can trigger on a condition found on one probe pod bit or bits followed_by a condition found on a different probe pod bit or bits.

**Halt_rept_exec duration is greater_than trace memory -** Displayed when the entered duration is greater than the duration of the trace memory.

**Halt_rept_exec duration is less_than sample period -** Displayed when the entered duration is less than the sample period.

**Hardware error in graphics printer board -** Displayed when the graphics printer board has a hardware malfunction.

**Hpib only possible in stand-alone mode -** Displayed when you try to specify the input or output device as an HP-IB port and you are operating in a cluster environment.

**IMB execution error -** Displayed when measurement_system module cannot execute all previously configured modules.

**Label table is full -** Displayed when you try to define more labels than the analyzer can use. The analyzer can operate with up to 31 labels defined (including current labels and compare file labels).

**Label width exceeds instrument's capabilities -** Displayed when you try to create a label having more probe bits than are available in the configuration of your analyzer.

**Labels used in any other specification cannot be deleted -** Displayed when you try to delete a label from this specification that is used as a parameter in some other specification in the analyzer.

**Lower threshold must be less than upper threshold -** Displayed during Dual_Threshold mode when you try to enter a specification which places the lower threshold at a voltage more positive than the upper threshold.

**Mark "after condition" not allowed -** Displayed when you try to mark a condition and you specify an "after" condition using the same mark ("mark x ... after mark_x" is invalid).

**Mark "before condition" not allowed -** Displayed when you try to mark a condition and you specify a "before" condition using the same mark ("mark x ... before mark_x" is invalid).

**Marking complete, marking limit of 511 exceeded -** Displayed when the analyzer finishes marking the data and has found more than 511 events that meet specifications of mark_<abcd>.

**Measurement in process, cannot alter specification -** Displayed when you try to modify any of the measurement parameters while the analyzer is involved in making a measurement.

**Measurement_system error -** Displayed when the measurement system software is not present.

**Mode is not glitch capture -** Displayed when you try to enter a trigger specification requiring glitch detection and the analyzer is not in the glitch mode of measurement.

**Module not involved -** Displayed when the analyzer module is not involved in the current measurement.

**Multiple IMB drivers -** Displayed when multiple drives are set up to operate on a given intermodule bus line.

**New label already exists -** Displayed when you try to rename a label to a label which already exists.

**"or" is not possible, all bits are already specified -** Displayed when you try to enter your trigger specification which requires a value to be found on a probe input and glitch detection on two or more probe pod inputs. Triggering can be specified to occur on a value ANDed with glitch detection on one probe input channel.

**"or" is not possible, all pods are already used -** Displayed when your trigger specification is an ORing of values in which at least one of the pods is mentioned more than one time. An "OR" condition can only occur between pods.

**"or_on" is not possible, all bits are already specified -** Displayed when you try to enter a trigger specification which calls for finding a value on a bit or set of bits when your specification already includes values for all the bits under other labels.

**Pod hardware is not present -** Displayed when you try to enter a specification for a condition to be found on a pod which does not exist in your configuration.

**Processing complete, marking limit of 511 exceeded -** Displayed when the analyzer completes processing the trace list and more than 511 mark_<abcd> events were found.

**Process_for_data definition is invalid -** Displayed when you change the mode or redefine the labels in such a way that the process_for_data definition is no longer valid.

**Range must be greater than one sample -** Displayed when you try to find an event and the range you specified for the analyzer to look in is only one sample wide.

**Sample exceeds memory depth -** Displayed when you enter a specification to process_for_data a number of samples "before" or "after" a transition and the number of samples is greater than the memory depth of the analyzer (4060 or 8140 samples).

**Sample period is fixed at 2.5 nsec in fast sample mode -** Displayed when you try to enter a different sample period specification while the analyzer is operating in the Fast Sample mode.

**Sample rate is fixed at 400 MHz in fast sample mode -** Displayed when you try to enter a different sample rate specification while the analyzer is operating in the Fast Sample mode.

**Sample rate is too slow for sampled trigger duration -** Displayed when you try to enter a combinational glitch trigger definition where the sample period is longer than the trigger duration. Glitch triggering requires all data to be sampled before trigger duration detection circuitry.

**Sequence limit is four marks -** Displayed when you try to enter a halt_repetitive_execution sequence which is greater than four marks long.

**Single bit label must be entered -** Displayed when you try to process_for_data relative to a transition on a label, and the label is defined as more than one bit.

**Single bit label must be entered to use "with" -** Displayed when you try to mark or find a value with a glitch and the label is wider than one bit.

**Specification does not agree with captured data -** Displayed when you command the analyzer to save its configuration with_data, after you have changed the specification to something different from what was used to capture the data.

**Support for function is not present in hardware -** Displayed when you try to drive or receive IMB signals which are not supported by another module.

**Trace complete -** Displayed when the analyzer has completed its trace in process.

**Trace halted -** Displayed when you halt the analyzer trace in process.

**Trace in process -** Displayed when the analyzer has found the condition that meets its trigger specification and is in the process of completing its trace.

**Trigger and trigger enable cannot both be received -** Displayed when you try to enter a specification that requires trigger and trigger enable both to be received from an associated analysis module. Either one can be received, but the analyzer cannot receive both.

**Trigger cannot be received or driven with restart function -** Displayed when you try to enter the trigger enable restart command and a trigger specification of "trigger driven" or "trigger received" exists.

**Trigger not found -** Displayed when the "hold" signal is received (IMB trigger received) before the analyzer has found its timing analysis trigger.

**Trigger pattern resource is used in trigger definition -** Displayed when you try to enter a trigger enable definition using the pattern resource. Must have "trigger received" before a trigger enable definition using the pattern resource is valid.

**Trigger pattern resource is used in trigger enable definition -** Displayed when you try to enter a trigger definition using the pattern resource which is already used in the trigger enable definition. Change the trigger enable component to not include the pattern resource and then reenter your trigger definition.

**Waiting for enable -** Displayed when the analyzer is searching for the pattern to drive the trigger enable line.

**Waiting for hold -** Displayed when the analyzer is waiting to receive the "hold" signal (IMB trigger) from another analyzer to restart its measurement.

**Waiting for trigger -** Displayed when the analyzer has satisfied its prestore requirements but has not found the specified trigger condition. Note: the external enable must be true before trigger will be found.

**"when_greater_than" is not possible, all pods are used -** Displayed when your trigger specification requires a condition to be found ANDed with a duration of a value to be found, and both conditions include bits from the same probe pod.

**"with" is not possible, need a single bit glitch label -** Displayed when you try to enter a trigger specification which requires a value to be found on a probe input and glitch detection on two or more probe pod inputs. Triggering can be specified to occur on a value ANDed with glitch detection on one probe input channel.

**-10.0 volts <= threshold <= 10.0 volts -** Displayed when you try to specify a threshold voltage that is outside the range of threshold voltages that the analyzer can accept.

**"<>" is not possible with "any_glitch on" -** Displayed when your trigger specification ANDs glitch detection with a "not-equal-to" value found on a label. Glitch detection can only be ANDed with "equal-to" values on labels.

# Index

64610-90901,October 1984


PRINTED IN U.S.A.