μAnalyst 2000
MODEL 2100

INTERACTIVE STATE ANALYZER

Operator's Manual

NWS

NORTHWEST INSTRUMENT SYSTEMS, INC.

████ WARRANTY

Northwest Instrument Systems, Inc., warrants this product against defective material and workmanship, under normal use, for a period of one year from the date of sale. At the option of Northwest Instrument Systems, Inc., proven defects covered by this warranty will be corrected either by repair or replacement.

This warranty applies only to products manufactured by Northwest Instrument Systems, Inc. This warranty does not apply if the product has been modified, altered, or serviced by other than Northwest Instrument Systems, Inc., or its authorized service representative. Furthermore, this warranty does not apply if the product has been damaged by misuse, abuse of any kind, misapplication, or accident.

If warranty repair is required, call one of the authorized service representatives listed in the *User Update Policy* at the front of this manual. Or, call Northwest Instrument Systems, Inc., (1-800-547-4445) for assistance in locating the authorized service center nearest you.

Service will not be performed without proof of purchase showing purchase date. All instruments must be securely packaged in their original packing material, or the equivalent, to prevent damage during shipment.

THERE ARE NO OTHER WARRANTIES, STATEMENTS OR REPRESENTATIONS, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, UNLESS STATED HEREIN. NORTHWEST INSTRUMENT SYSTEMS, INC., SHALL NOT BE LIABLE FOR ANY INCIDENTAL, CONSEQUENTIAL, OR DIRECT DAMAGE RESULTING FROM THE USE OR MISUSE OF THIS PRODUCT; OR RESULTING FROM THE BREACH OF ANY EXPRESS OR IMPLIED WARRANTY, INCLUDING DAMAGE TO PROPERTY.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

■■■■ Electromagnetic Emissions

The μAnalyst® 2000 system has been designed to minimize emitted and conducted electromagnetic energy. Tests have shown that when properly installed with an IBM® or IBM-compatible personal computer that meets FCC Class B emission standards, the μAnalyst system meets or exceeds applicable standards.

Each owner is subject to the non-interference requirements of Section 15.803 of the FCC Rules. In effect, Section 15.803 states that the operator of a computing device may be required to stop operating his device if it is found that the device is causing harmful interference and it is in the public interest to stop operation. Operation may not resume until the interference problem is corrected.

μAnalyst 2000
MODEL 2100

First Edition: January, 1985

Second Edition: January, 1986

iv

■■■ READ THIS FIRST

In this manual you'll find the information you need to operate the Northwest Instrument Systems, Inc., Model 2100 Interactive State Analyzer, as part of the μAnalyst® 2000 instrumentation system.

The manual explains the features of the Interactive State Analyzer and describes all operating menus and displays. It also shows how to use the Interactive State Analyzer in time-aligned acquisitions with the Model 2200 Interactive Timing Analyzer.

A special feature of the manual is the demonstration session in Section 7. This session gives you the opportunity to operate the Interactive State Analyzer's hardware and software, before getting into actual applications.

Take a few minutes to look over the following pages. They describe how the manual is organized and explain which product and software versions are covered.

■■■ USING THIS MANUAL

This manual has seven sections and several appendices.

Introduction	Section 1 provides an overview of the Interactive State Analyzer and its key features. It also lists the analyzer's components and software specifications.
Installation	Installation information is now described in a separate document, titled *Model 2100 Interactive State Analyzer Hardware Installation Manual Insert*.
Getting Started	Section 3 shows how to use the μAnalyst software with floppy and fixed-disk computer systems. It also explains the μAnalyst's start-up procedures and defines any possible error conditions.
Modes of Operation	Section 4 discusses the Interactive State Analyzer's key features and their theory of operation.
Menus	Section 5 covers all operational and mechanical aspects of the Interactive State Analyzer menus.

State & Timing Together	Section 6 describes how to use the Interactive State Analyzer in time-aligned acquisition with the Model 2200 Interactive Timing Analyzer.
Demonstration	Section 7 provides hands-on exercises that use the Interactive State Analyzer to acquire data from a demo circuit board.
Appendices	Appendices at the back of the manual provide reference information, such as data file formats and clocking examples.

Each section in the manual is preceded by a tabbed page so that information can be referenced easily. Other reference aids include a table of contents and a full cross-reference index.

■ μAnalyst 2000
MODEL 2100

■ PRODUCTS COVERED BY THIS MANUAL

The information in this manual applies to the μAnalyst products listed below.

μAnalyst 2000 Mainframe

Model 2100(E) Interactive State Analyzer (ISA)

ISA Controller Board

ISA 16-Channel Memory Board (Option 001)

ISA 32-Channel Memory Board (Option 002)

Procedures for installing the μAnalyst 2000 Mainframe are provided in the *Mainframe Installation* section of the *μAnalyst 2000 Hardware Installation Manual*.

Procedures for installing the Model 2100 Interactive State Analyzer instrument boards are provided in the *Model 2100 Interactive State Analyzer Hardware Installation Manual Insert*, shipped with the state analyzer.

■■■■ SOFTWARE VERSIONS COVERED BY THIS MANUAL

The information in this manual applies to the SW2000 μAnalyst State/Timing Analysis Software. This software is comprised of the two diskettes listed below.

SYSTEM DISK V1.00 and later, DOS-Compatible

HELP DISK V1.00 and later, DOS-Compatible

The SYSTEM DISK is available in two versions: 512K or 256K. The 512K version can be used with computers having 512K or greater memory, while the 256K version can be used with computers having 256K or greater memory. When you are using the 256K version, the size of the state analyzer's reference memory is limited to the amount of free space in RAM. To fully utilize the reference memory feature of the state analyzer, 320K of RAM is required. If less than 320K of RAM is resident in the computer, the reference memory can hold about 1K of a 4K data acquisition.

Only one SYSTEM DISK is needed to operate the μAnalyst. Use the disk that is appropriate for the memory capacity of your personal computer.

Procedures for loading the μAnalyst's software are provided in the *Getting Started* section of this manual.

■■■ RELATED PRODUCTS AND OPTIONS

The following products and options may be operated in conjunction with the Model 2100 Interactive State Analyzer.

Model 2200 Interactive Timing Analyzer

The Model 2200 is a 16-channel, 100 MHz timing analyzer. You can operate this product independently within the mainframe, or you can use it in time-aligned acquisition with the Interactive State Analyzer.

The Model 2200 comes equipped with a memory board, a trigger board, three data probes, and a user's manual.

Time Stamp Board (Option 003)

This board is available as an option to the Interactive State Analyzer. It provides 16 input channels, coupled with a 20 MHz asynchronous clock. You can measure the real-time execution of state events and view their time distributions in a histogram format.

The Time Stamp comes equipped with a memory board, a 16-channel data probe, and a manual supplement.

Mnemonic Disassemblers

A variety of mnemonic disassemblers are available for the Interactive State Analyzer, including:

Z80

8085

8086

8088

68000

68008

68010

Each disassembler comes equipped with mnemonics software, a single-plug probe connector, and a manual supplement.

New disassembler packages are currently under development. For more information, contact Northwest Instrument Systems, Inc.

■■■ PC COMPATIBILITY REQUIREMENTS

To operate the *µ*Analyst products and software covered in this manual, you must have a personal computer that meets the requirements listed below.

PC Compatibility Requirements

Type	IBM® PC™, PC AT™, PC XT™; or COMPAQ™, COMPAQ PLUS™
Drives	2 double-sided disk drives; or 1 fixed disk, plus 1 double-sided disk drive.
Memory	256K or 512K RAM.
*Monitor	IBM Monochrome Display and Printer Adapter; or IBM Color Display and Color/Graphics Monitor Adapter.
Software	Disk Operating System (DOS) Version 2.00 or higher.

*The IBM Color Display and Color/Graphics Monitor Adapter must be used if you are operating the Model 2200 Interactive Timing Analyzer.

If desired, you may also use a printer in conjunction with the *µ*Analyst. Compatible printers include:

■ Epson RX-80, MX-80, FX-80

■ IBM PC Graphics Printer

■■■ CONTENTS

Section 1: Introduction

Section 2: Installation

Section 3: Getting Started

Section 4: Modes of Operation

Section 5: Menus

Section 5: Menus (continued)

Section 5: Menus (continued)

Section 6: State & Timing Together

Section 7: Demonstration

████ LIST OF ILLUSTRATIONS

████ LIST OF TABLES

Using this section. This section introduces you to the μAnalyst 2000 and its Model 2100 Interactive State Analyzer (ISA). It describes the key product features, components, and specifications.

■ SECTION CONTENTS

■■■ DESCRIPTION

The Model 2100 Interactive State Analyzer (ISA) is a powerful tool for designing, debugging, and evaluating software-intensive systems. It installs in the μAnalyst 2000 Mainframe and operates under the control of your IBM-compatible personal computer.

The state analyzer's hardware consists of a controller board and expandable memory boards. These boards plug into the expansion slots located within the μAnalyst mainframe chassis. You can configure the state analyzer with 16 to 80 acquisition channels, using any combination of the following board types:

■ Controller Board (one required)—provides synchronous clocking, triggering, and control logic.

■ 16-Channel Memory Board (optional)—provides 16 acquisition channels with 4096 bits-per-channel memory.

■ 32-Channel Memory Board (optional)—provides 32 acquisition channels with 4096 bits-per-channel memory.

A minimum state analyzer configuration consists of the controller board and at least one memory board.

Once installed in the mainframe, the state analyzer is controlled by menu-driven operating software. This software is contained on the μAnalyst's SYSTEM and HELP DISKS.

The various software menus are both easy to learn and use. They control all facets of the analyzer's operation, including clocking, triggering, storage qualification, and data display. They also control linked operations between the state analyzer and other μAnalyst products, such as the Model 2200 Interactive Timing Analyzer.

The rest of this section discusses the state analyzer's key features and system components. It also gives you a complete list of hardware specifications and software characteristics.

████ KEY FEATURES

Key features of the Interactive State Analyzer include:

■ **Universal Clocking**—Five external sample clocks make the state analyzer compatible with virtually any type of microprocessor system. You can use one or all of these sample clocks to synthesize a master sampling clock. The clock lines are logically ORed, and they each provide a data sampling rate up to 10 MHz.

■ **Single-Probe Demultiplexing**—Two hold clocks let you demultiplex data without double probing. You can use these clocks to latch and hold data until the master sampling clock is valid. The demultiplexing is all handled internally by the state analyzer.

■ **15-Level Triggering and Storage Qualification**—You have up to 15 trigger/store states at your disposal when defining a trigger. Each state contains four word recognizers, which can be arranged in powerful IF-THEN-ELSE sequences that perform such actions as activating the trigger, branching to a different state, or qualifying data for storage.

■ **Multiple-Preview Acquisition**—A special acquisition mode enables you to acquire a qualified event, as well as to capture the seven events directly preceding that event. This gives you all the benefits of storage qualification, plus the ability to trace information leading up to each qualified event.

■ **Acquisition and Reference Memories**—The state analyzer's acquisition and reference memories can be displayed in hexadecimal, octal, binary, or ASCII radices. You can view these two memories side by side, and make continuous acquisitions until both memories are equal or not equal. The differences between the two memories are highlighted on the display screen.

■ **Histogram Overview Displays**—The state analyzer's histogram display lets you view an overall picture of system activity. In this display, acquired events are binned into user-defined ranges, and you see a histogram readout of the percentage of activity falling within each range.

- **Time Stamp Measurements**—You can add a Time Stamp Board to the state analyzer at any time. This board provides a 20 MHz asynchronous clock that lets you measure the real-time execution of your software code. The time distribution of recorded events is shown in the state and histogram displays. (See *Related Products and Options* at the front of this manual.)

- **Mnemonic Disassemblers**—The state analyzer offers a wide variety of mnemonic disassemblers which can convert acquired microprocessor software back into an assembly source listing format. Each disassembler comes equipped with its own disassembly software and a single-plug probe connector. (See *Related Products and Options* at the front of this manual.)

- **Time-Aligned Displays with Timing Analyzer**—The state analyzer can share the μAnalyst mainframe with the Model 2200 Interactive Timing Analyzer (see Section 6 of this manual). A cross-aligned mode lets you operate the two analyzers in tandem, where state events arm timing acquisitions, or vice versa. The state and timing data is then time-correlated on the display screen.

- **I/O Utilities**—Mass storage is available through the state analyzer's interface to the personal computer. You can save menu setups and reference data on disk, then recall them at any time. You can also make permanent records of setups and data by using an IBM-compatible printer.

SYSTEM COMPONENTS

The following paragraphs briefly describe the components required for operating the Model 2100 Interactive State Analyzer. While reading these paragraphs, refer to Figure 1-1 for an overview of how these components fit together.

☑ **NOTE**

Installation procedures for the hardware components are provided in the *Model 2100 Interactive State Analyzer Hardware Installation Manual Insert.*

State Analyzer Boards and Probes

The state analyzer boards plug into the expansion slots located within the μAnalyst 2000 Mainframe. Probes are connected to these boards through slot openings in the mainframe's back panel.

The state analyzer boards and probes include:

■ Controller Board, with one P2901 Clock Probe

■ 16-Channel Memory Board, with one P2902 Data Probe

■ 32-Channel Memory Board, with two P2902 Data Probes

A minimum state analyzer configuration requires the controller board and at least one memory board. A maximum configuration allows any number of memory boards up to 80 parallel channels.

μAnalyst 2000 Mainframe

The μAnalyst 2000 Mainframe houses the state analyzer boards, and provides the hardware interface to the personal computer. This interface consists of:

■ μAnalyst Interface Board

■ PC Interface Board

■ 48-inch Interface Cable

☞ NOTE
The mainframe can house other μAnalyst products simultaneously with the state analyzer. Refer to *Related Products and Options* at the front of this manual.

μAnalyst Software

The μAnalyst System Operating Software is provided as part of the state analyzer package. It contains the menus necessary for operating the analyzer by itself or in conjunction with other μAnalyst products.

PC Compatibility

An IBM or IBM-compatible personal computer, and DOS operating software, are essential to the state analyzer and all μAnalyst products. A list of compatible personal computers and their memory requirements are provided at the front of this manual under *PC Compatibility Requirements*.

■ **Table 1-1 (cont.): Software Characteristics**

■ Symbol Menu: word recognizer symbol name and value definition; mask definition.

■ Display Menu: hexadecimal, octal, binary, and ASCII display formats; data display of acquisition, reference, or both; display of memory comparisons with data differences highlighted; single line or page scrolling; find word (scroll by value); compress mode.

■ Histogram Menu: 16 user-selectable ranges; 1 qualifier event; activity percentages shown in histogram and numerical readout.

■ I/O Menu: save and load setup and data to and from disk; path name setup; disk directory display; print data, symbols, or trigger information.

■ Configuration Menu: μAnalyst mainframe hardware description; interface address selection; software option selections.

■ Help Information: general help summary; individual help summary for menu fields.

Run (Go) Modes

Go once; Go forever; Go until acquisition = reference; Go until acquisition < > reference; Stop acquisition after memory overflow.

■■■■ ERRATA INFORMATION

Installation information is now described in a separate document titled *Model 2100 Interactive State Analyzer Hardware Installation Manual Insert.*

This insert was shipped with your state analyzer with the instructions to insert it into the *Instrument Installation* section of the *μAnalyst 2000 Hardware Installation Manual.*

Using this section. This section shows you how to load the μAnalyst software and start the system. It also provides instructions on how to run the μAnalyst software from a fixed disk.

■■■ SECTION CONTENTS

THE μANALYST SOFTWARE

The μAnalyst's software package contains three diskettes:

- **SYSTEM DISK for 512K**—runs on computers that have a memory capacity of 512K or greater.

- **SYSTEM DISK for 256K**—runs on computers that have a memory capacity of 256K or greater.

- **HELP DISK**—runs on computers that have a memory capacity of 256K or greater.

Each SYSTEM DISK contains the entire μAnalyst menu operating software. Only one is required per system. Use the SYSTEM DISK that fits the memory capacity of your personal computer.

☞ **NOTE**
The SYSTEM DISK for 256K memory has certain limitations. For details, refer to the front of this manual, under the page titled *Software Versions Covered in this Manual.*

The HELP DISK can be used with either 512K or 256K of computer memory. This diskette is not required for operating the μAnalyst, but it is recommened. It provides a Help Information System that is comprehensive and extremely useful. In addition, it provides demonstration and training programs that will assist you in learning the state analyzer's features and operations (see Section 7 of this manual).

The rest of this section shows you how to load and operate the SYSTEM and HELP DISKS.

USING DISKETTES
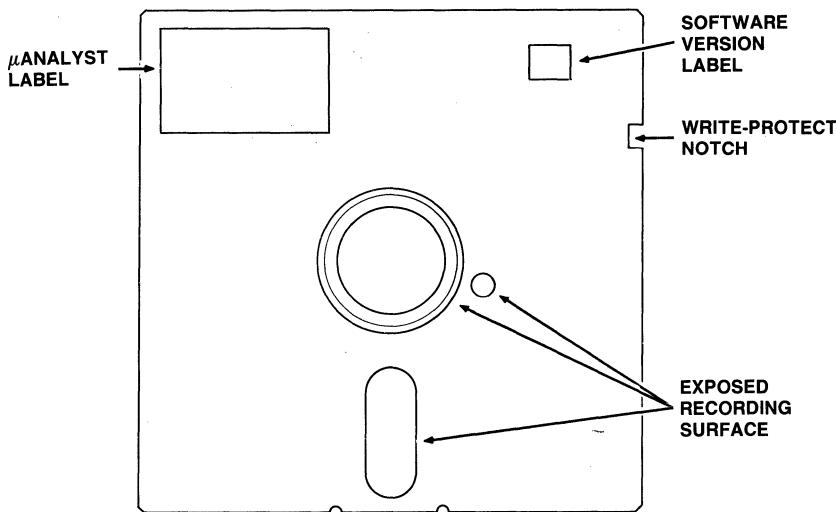
Diskettes require special care in handling and storage. When using diskettes, you should always observe the following rules:

- Hold diskettes by their upper left or right corners. Never touch their exposed recording surfaces (see Figure 3-1).

- Never fold or bend diskettes.

- When diskettes are not in use, store them in their accompanying paper envelopes.

- Store diskettes in a dry area with temperatures between 10°-52° C (50°-125° F).

- Never expose diskettes to magnetic fields, such as those surrounding permanent magnets or electronic motors.

00842-016

■ **Figure 3-1. A μAnalyst diskette.**

████ COPYING DISKETTES FOR BACKUP

When you first unpack the μAnalyst diskettes, place adhesive tabs over their write-protect notches. Then use the diskettes to make back-up copies.

The procedures for making back-up copies are listed below. These procedures assume that you know how to use the DISKCOPY command. If not, refer to your DOS reference manual.

To make back-up copies:

1. Install a bootable DOS diskette (version 2.00 or higher) into drive A of the personal computer, then turn the computer on.

2. Now, use the DISKCOPY command to copy the files from the μAnalyst's SYSTEM DISK (source diskette) to a new, blank diskette (target diskette).

3. Finally, use the DISKCOPY command to copy the files from the μAnalyst's HELP DISK (source diskette) to the another new, blank diskette (target diskette).

☑ **NOTE**

The DISKCOPY command automatically formats the target diskettes when it copies the files.

Continue using the above procedures to make as many back-up copies as you want. When you're finished, put the write-protected diskettes in a safe place, and use the back-up copies for operating the μAnalyst.

■■■ USING A FIXED DISK

As an alternative to using diskettes, you can install the μAnalyst software in a fixed disk. This fixed disk must contain the DOS operating system (version 2.00 or higher), and it must have at least 500K bytes of free disk space.

☑ **NOTE**

For a complete list of compatible fixed-disk computer systems, refer to the page titled *PC Compatibility Requirements* at the front of this manual.

You install the μAnalyst software in a fixed disk by executing the FINSTALL batch file as described below. This batch file is contained in the μAnalyst's SYSTEM DISK. When it is executed, the batch file creates a subdirectory titled ANALYST at the root level on the fixed disk. It then copies all of the μAnalyst system and help files into this subdirectory.

To use the FINSTALL batch file:

1. Power up your fixed-disk computer, and if necessary, load the DOS operating system.

2. Insert the μAnalyst's SYSTEM DISK into drive A (the floppy disk drive of the fixed-disk computer).

3. Enter drive A.

4. Type the batch file command:

    FINSTALL C: (where C is the fixed-disk drive of your computer.)

    Then press the return key.

5. Follow the on-screen instructions provided by the batch file. A message on the screen will tell you when the copying procedure is complete.

████ STARTING THE μANALYST

☑ **NOTE**
The following procedures assume that you have installed the
μAnalyst hardware and have connected the mainframe to the per-
sonal computer as specified in Section 2 of this manual. If not, do
so at this time.

There are two methods of starting the μAnalyst: one for
diskettes and one for fixed disks. These two methods are described
in the following paragraphs.

Starting the μAnalyst Using Diskettes:

1. Insert a bootable DOS diskette (version 2.00 or higher) into
drive A of the personal computer.

2. Turn on both the personal computer and the μAnalyst
mainframe.

3. Remove the DOS diskette from drive A and replace it with
the μAnalyst's HELP DISK.

4. Now, insert the μAnalyst's SYSTEM DISK into drive B of the
personal computer.

5. Enter drive B. Then type the word ANALYZE and press the
return key.

The start-up procedure is now complete.

Starting the μAnalyst Using a Fixed Disk:

☑ **NOTE**
The following procedures assume you have transferred the μAnalyst
software onto a fixed-disk computer as described earlier under
Using a Fixed Disk.

① Turn on both the fixed-disk computer and the μAnalyst.

② Enter drive C (the fixed-disk drive).

③ Enter the ANALYST subdirectory by typing the Change
Directory command as follows:

          CD ANALYST

Then press the return key.

④ Finally, type the word ANALYZE and press the return key.
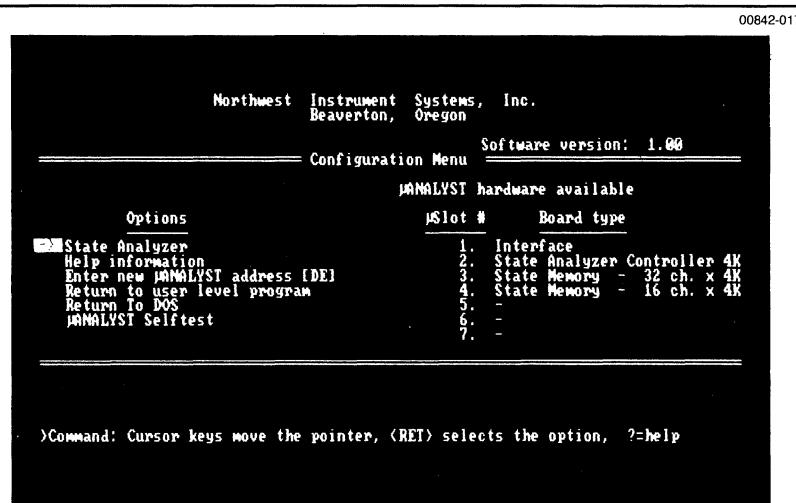
The start-up procedure is now complete.

A Successful Start-Up Display

The Configuration Menu is the first menu displayed by the
μAnalyst. This menu tells you the status of the μAnalyst hardware and
software, and it directs you to the various levels of operation.

Figure 3-2 illustrates a successful start-up display of the Con-
figuration Menu. The right side of menu tells you the status of the
μAnalyst. It specifies the software version in use, and it lists all prod-
uct boards and their mainframe slot locations.

The left side of the menu provides a list of the μAnalyst's
operating options. Through these options, you access the various
menus and control all system-level functions. *(Note: We will describe
these options in detail in Section 5 of this manual.)*

00842-017


```
                    Northwest  Instrument  Systems,  Inc.
                         Beaverton,  Oregon
                                                Software version:  1.00
                              Configuration Menu

                                   µANALYST  hardware  available

              Options                   µSlot #      Board type
        State Analyzer                     1.   Interface
         Help information                  2.   State Analyzer Controller 4K
         Enter new µANALYST address [DE]   3.   State Memory  -  32 ch. x 4K
         Return to user level program      4.   State Memory  -  16 ch. x 4K
         Return To DOS                     5.   -
         µANALYST Selftest                 6.   -
                                           7.   -




        >Command: Cursor keys move the pointer, <RET> selects the option,  ?=help
```

■ **Figure 3-2. A successful start-up display.**

Start-Up Error Conditions

When the μAnalyst software is first loaded, it tests the PC
interface address and checks the μAnalyst mainframe for installed cir-
cuit boards. The following paragraphs list and define the possible
start-up error conditions.

**Error Condition: The Configuration Menu does not appear on
start-up.**
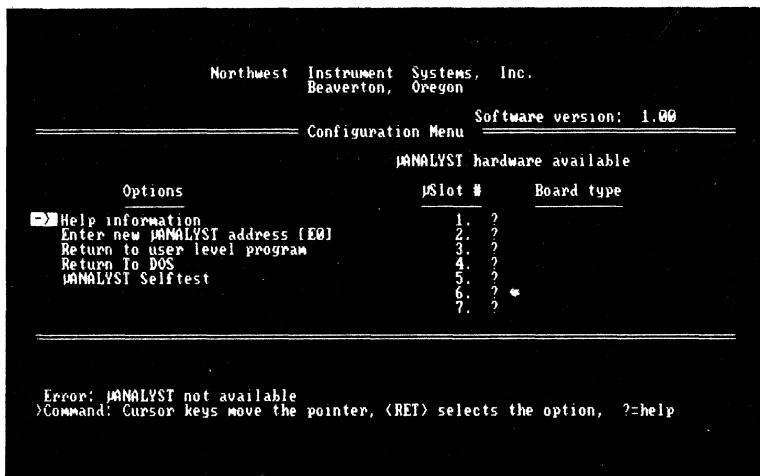
This error indicates a problem with your personal computer or with the μAnalyst SYSTEM DISK. First, check your personal computer to see if it is working properly. Then, check the label on your μAnalyst SYSTEM DISK to make sure it fits the memory capacity of your computer.

If the above two procedures do not locate the problem, make a new back-up copy from your original μAnalyst SYSTEM DISK and try reloading the software. If the new back-up copy fails, contact Northwest Instrument Systems, Inc.

Error Condition: The Configuration Menu displays a question mark (?) next to each of the slot locations (see Figure 3-3).

<div style="text-align:right">00842-018</div>

```
                    Northwest  Instrument  Systems,  Inc.
                    Beaverton,  Oregon

                                           Software version:  1.00
    ========================= Configuration Menu =================

                                 μANALYST hardware available

            Options                 μSlot #       Board type
    ----------                      ------        ----------
  ▶ Help information                   1.  ?
    Enter new μANALYST address [E0]    2.  ?
    Return to user level program      3.  ?
    Return To DOS                      4.  ?
    μANALYST Selftest                  5.  ?
                                       6.  ?  *
                                       7.  ?



    Error: μANALYST not available
    >Command: Cursor keys move the pointer, <RET> selects the option,  ?=help
```

■ **Figure 3-3. Start-up error condition.**

If your display looks like Figure 3-3, one of five things is probably wrong:
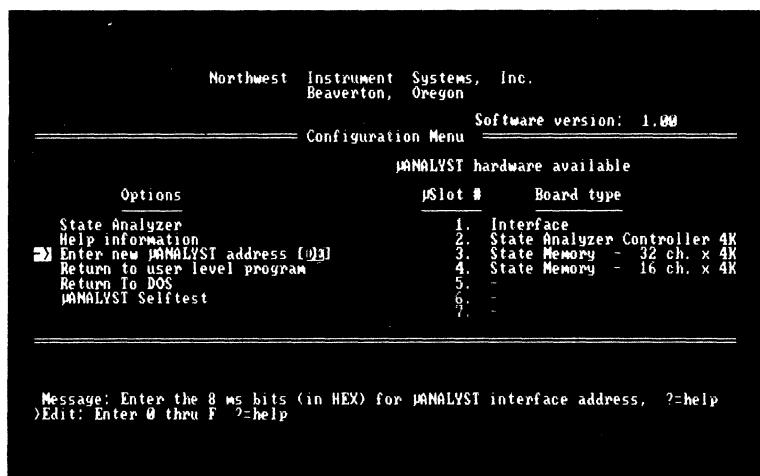
1. The interface cable between the μAnalyst and the personal computer is not securely connected.

2. The power to the μAnalyst mainframe is not on.

3. The μAnalyst or PC interface boards are poorly seated in their slot connectors.

4. The address switches on the PC interface board have been set to an address range that is being used by another PC-based product.

5. The address switches on the PC interface board have been changed from their original default setting of DE hexadecimal.

If the error is caused by one of the first three problems, take appropriate action.

If the error is caused by the fourth problem, you will have to change the address range for either the μAnalyst or the conflicting product. Procedures for changing the μAnalyst's address range and it switch settings are provided in Section 2 of this manual, under the subsection titled *Installing the PC Interface Board*.

00842-019


```
                    Northwest  Instrument  Systems,  Inc.
                            Beaverton,  Oregon
                                          Software version:  1.00
    ============================ Configuration Menu ============================

                                    μANALYST hardware available

            Options                    μSlot #      Board type

    State Analyzer                       1.   Interface
    Help information                     2.   State Analyzer Controller 4K
 ➤  Enter new μANALYST address [0]3]     3.   State Memory  -  32 ch. x 4K
    Return to user level program         4.   State Memory  -  16 ch. x 4K
    Return To DOS                        5.   -
    μANALYST Selftest                    6.   -
                                         7.   -

    ============================================================================


    Message: Enter the 8 ms bits (in HEX) for μANALYST interface address,  ?=help
    )Edit: Enter 0 thru F  ?=help
```

■ **Figure 3-4. Entering a new μAnalyst address.**

If the error is caused by the fifth problem, you can rectify the condition by entering a new address range into the Configuration Menu. To do this, move the blinking screen pointer to the Configuration Menu's Enter New μAnalyst Address option selection as shown in Figure 3-4. Press the return key. Now, using hexadecimal notation, enter the new address range into the reverse video field. Again, press the return key.

The Configuration Menu will check the newly entered address location, and if it finds the µAnalyst software, it will immediately replace the question marks with the appropriate hardware list.

☑ **NOTE**

If the SYSTEM DISK is not write-protected, the Configuration Menu will remember the newly entered µAnalyst address and will automatically search that location whenever the µAnalyst is restarted.

If none of the above procedures corrects the error condition, contact Northwest Instrument Systems, Inc.

Error: The Configuration Menu displays a question mark (?) next to a specific slot location.

A question mark next a slot indicates that the board installed in that slot is either poorly seated or faulty. If the problem still exists after you have reseated the board, contact Northwest Instrument Systems, Inc.

Using this section. This section gives you an overview of the Inter-active State Analyzer and its operating features. As you use the analyzer, keep in mind the main points of this section. The text here will help you program the analyzer with more understanding, and you will find that desired results are easier to define and achieve.

Throughout this section, reference is made to the various state analyzer menus. The menus are described in detail in Section 5 of this manual.

■ SECTION CONTENTS

■■■■ DATA ACQUISITION MODES

The Interactive State Analyzer has two data acquisition modes: the standard mode, where data is sampled and stored according to the storage qualification established in the Trigger-Store Menu; and the Multiple-Preview Acquisition (MPA) mode, where a sequence of unqualified data occurring before each qualified event is also stored.

Standard Data Acquisition Mode

In the standard data acquisition mode, data is sampled at each sample clock cycle and compared to a set of word recognizer values specified in the Trigger-Store Menu. Qualified data (data words matching the events programmed in the storage qualification IF <event> THEN STORE line of each state) is stored when the comparison is valid. The trigger event is always stored, too.

As an option (in the Trigger-Store Environment Submenu), you can also program the analyzer to store each event that causes a state transition. Since state names are stored along with the event that caused the state transition, program flows are easily traced. The demonstration example in Section 7 uses this capability to follow the operation of the demo circuit board.

Multiple-Preview Acquisition Mode

Qualified data storage is a powerful feature that allows you to define what kind of data to store, and ignore the rest. Multiple-Preview Acquisition (MPA) extends this concept, enabling you to solve more complex problems.
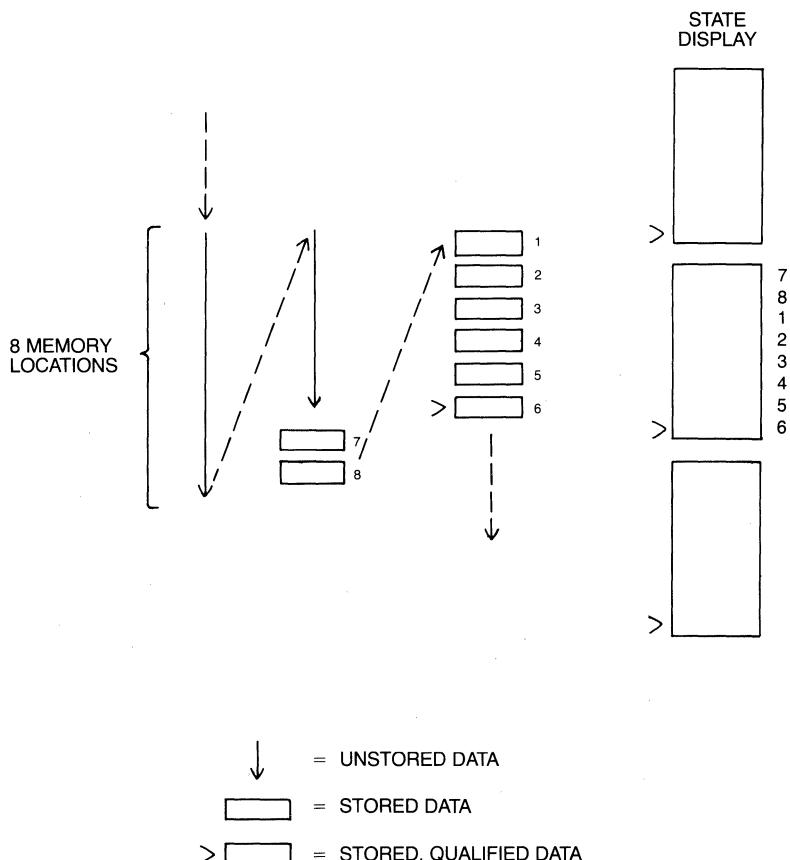
For example, suppose you know that a certain event is associated with a failure. You could try to find the cause of the failure by triggering on the failure-related event and looking at the data preceding it. But if the failure is intermittent, you would be forced to start the logic analyzer repeatedly until the cause was captured. MPA helps solve this kind of problem.

In MPA mode, the state analyzer's hardware is reconfigured so that data is stored continually while the machine waits for a qualified acquisition. When qualified data is detected, up to seven of the previous unqualified events are retained in memory, while the storage process begins again, until the system triggers. This process is illustrated in the example shown in Figure 4-1.

00842-020

STATE
DISPLAY

↓   = UNSTORED DATA

▭   = STORED DATA

> ▭   = STORED, QUALIFIED DATA

■ **Figure 4-1. MPA data acquisition.**
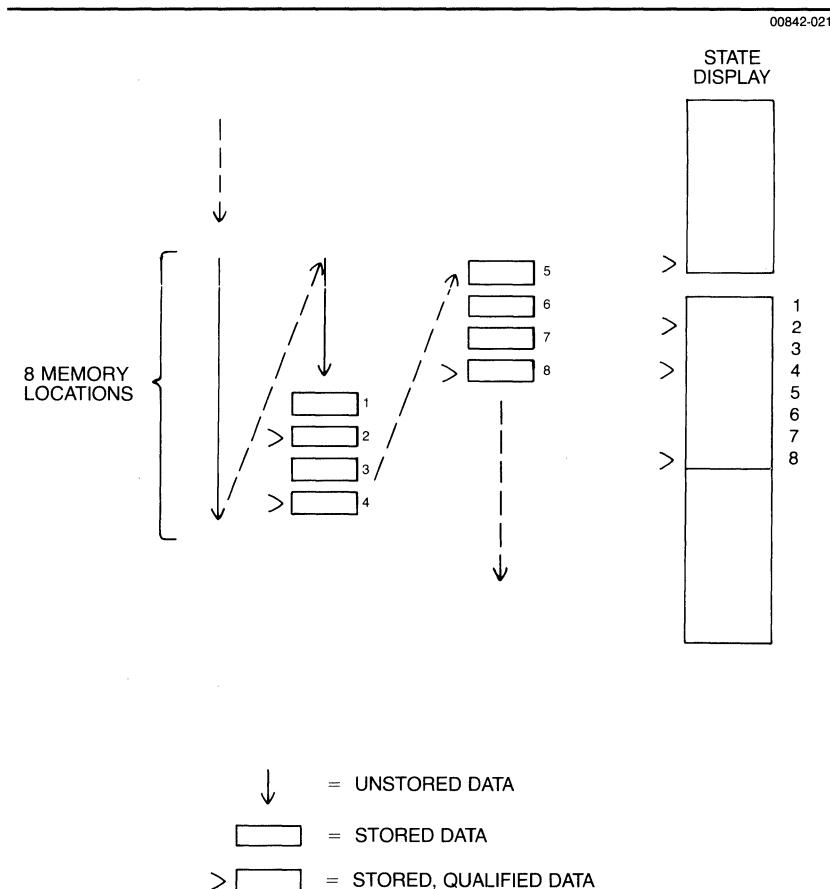
As shown in the example, unqualified data storage first fills location 1, then location 2, and so forth with each sample clock. If no qualified data is detected before location 8 has been written, new unqualified data overwrites location 1, then 2, and so on. In this particular example, a qualified event is finally detected and stored in location 6. Locations 7-8 and 1-5 now contain unqualified data stored on the seven sample clock cycles immediately preceding the qualified event.

On the state display, you will generally see data displayed as multiple groups of eight words. The qualified words (those matching the IF <event> THEN STORE line in the Trigger-Store Menu) are marked with a right arrow symbol (>).

In Figure 4-2 you can see that several qualified events occurred within eight sample clocks of one another. In this case, storage continues in the same group of eight locations, and then immediately continues in the next group of eight. In this example, the event in location 8 happened to be qualified.

00842-021

= UNSTORED DATA

= STORED DATA

= STORED, QUALIFIED DATA

■ **Figure 4-2. Qualified events, all within eight sample clocks.**
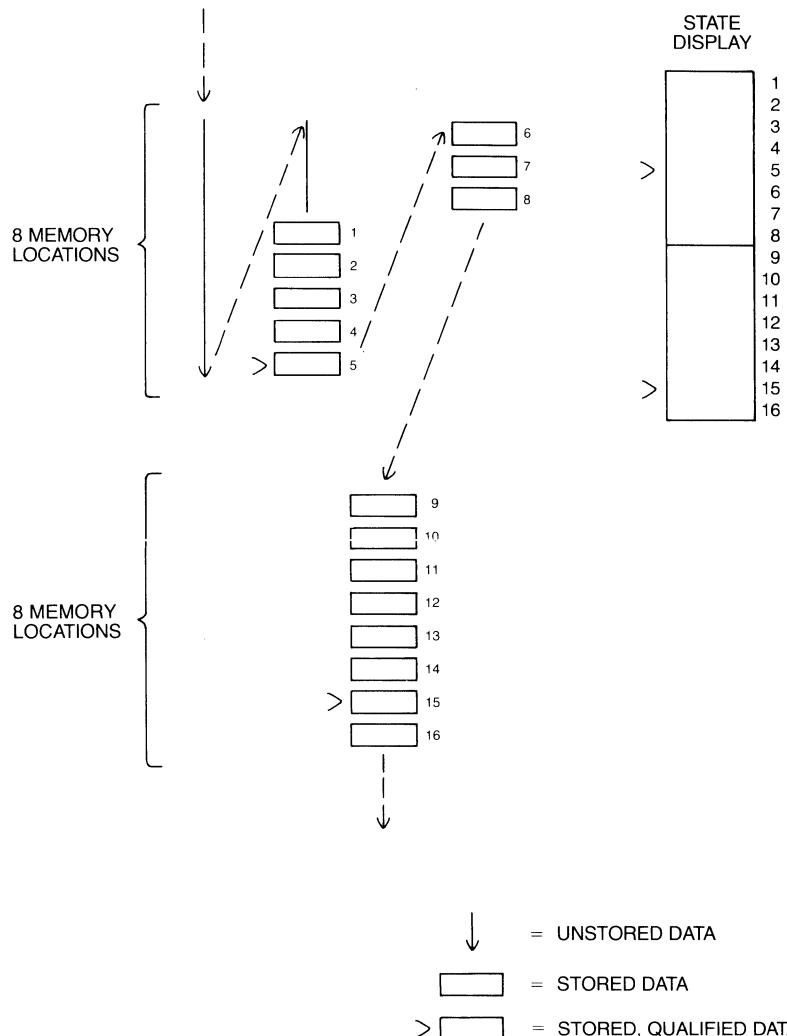
If the event in location 8 had not been qualified, the acquisition could have continued in much the same way as the acquisition represented in Figure 4-3. As in the example in Figure 4-2, several qualified acquisitions are made within eight sample clocks. When all eight locations in the current group are filled, new unqualified samples are stored in the next group.

If a qualified acquisition occurs before all the locations in this group of eight have had data stored in them once, this is treated as a special case in the state display. Note, in the state display to the right of Figure 4-3, that there are no blank lines between the two groups of eight unqualified events. Data not separated by blank lines is acquired on successive sample clocks, while a blank line between two lines of data indicates that an indeterminate number of sample clocks occurred between these two events.

00842-022

Figure 4-3. Unqualified events, within eight sample clocks.

As an example of how useful MPA mode can be, suppose you are debugging a program module you've just written. One bug seems to be associated with a global variable accessed by other modules you did not write. You could use the standard data acquisition mode to monitor the reading and writing of this variable by qualifying storage on the variable address, but this would not enable you to store the code that caused the variable to change. With the MPA mode, using the same storage qualification and triggering, the state analyzer will capture the executing code causing each of up to 512 accesses to this variable. Now you can search out the bug by looking back through the data acquired before the trigger.

☞ **NOTE**

For a more detailed example of how to use the MPA mode, refer to Section 7 of this manual.

■■■■ CLOCKING OPTIONS

The Interactive State Analyzer features a sophisticated clocking scheme. The P2901 Clock Probe provides seven input signals for use in establishing the synchronous sample and hold clocks. It also provides three output signals, the Run/Stop line and the Real-Time Programmable Outputs, which can be used for interacting with the system under test.

The following paragraphs describe how these input and output signals are used. The sample clocks, hold clocks, and Run/Stop signal are all programmed in the Format Menu. The Real-Time Programmable Outputs are programmed in the Trigger-Store Menu.

☞ **NOTE**

Detailed procedures for using the state analyzer clocking options to acquire data from various microprocessors are provided in Appendix C of this manual.

Sample Clocking

The sample clocks determine the time at which the state memory boards sample synchronous data. There are five sample clocks, which are ORed together to create the master clock. They are labeled S1 through S5 on the clock probe.

As you can see from the example in Figure 4-4, when the sample clocks do not overlap in time, they produce a master clock consisting of pulses contributed by each of the sample clock pulses.
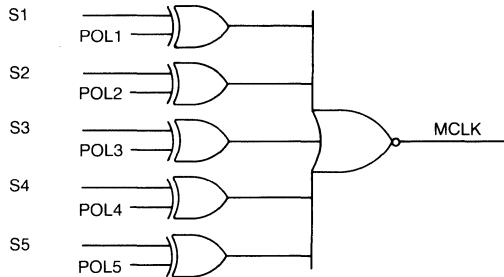
When the sample clocks do overlap, the master clock, which is the sum of all the sample clocks, merges the pulses to create a longer pulse.
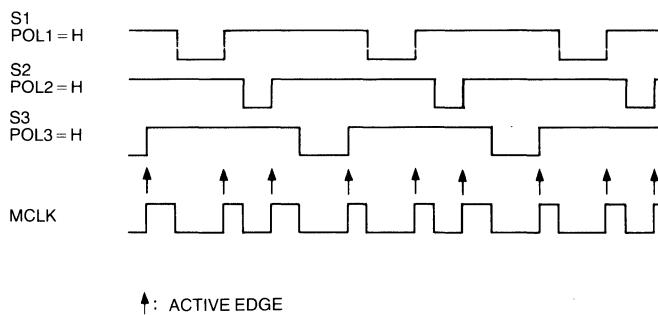
You select the polarity of each sample clock in the Format Menu. The slope referred to on the menu is the slope of the trailing edge of the sample clock—the edge on which data is sampled.
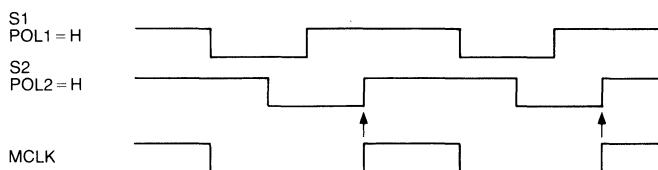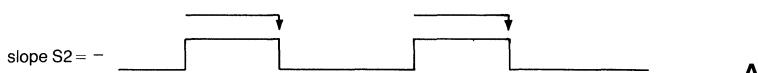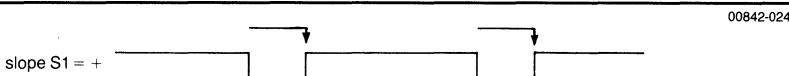
Figure 4-4. Using the five sample clocks to form the master clock.
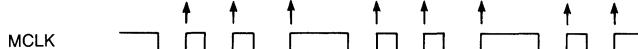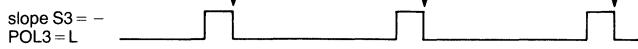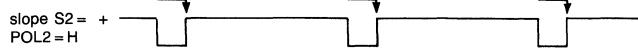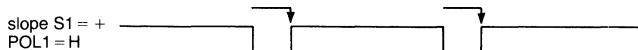
The polarity of the slope on the sample clocks affects the ORing of the clocks. Figure 4-5 shows how a simple change of polarity changes the master sample clock considerably. One way to look at this is in terms of clock periods. By defining the trailing edge of the clock, you are also defining the period from the leading to the trailing edge to be used in ORing the clocks together.

00842-024

slope S1 = +

slope S2 = −

A

Data is sampled at the point indicated by ↑

slope S1 = +
POL1 = H

slope S2 = +
POL2 = H

slope S3 = −
POL3 = L

MCLK

B

slope S1 = +
POL1 = H

slope S2 = −
POL2 = L

slope S3 = −
POL3 = L

MCLK
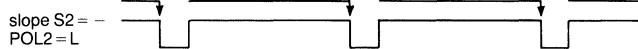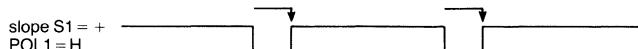
C
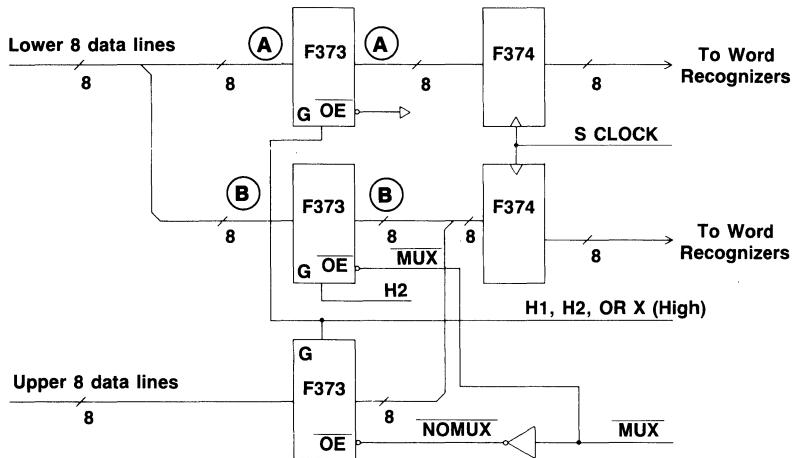
■ **Figure 4-5. The effects of sample clock polarity.**

Hold Clocks and Demultiplexing

For microprocessors or systems that multiplex data and addresses onto the same bus, the Interactive State Analyzer provides a convenient way to demultiplex the signals. Through the Format Menu, you can select multiplexed or non-multiplexed acquisition for each data probe.

Figure 4-6 illustrates how the data channels from a probe are routed to the word recognizers. Typically, all 16 channels are routed via Path A. In the multiplex mode, however, the probe's upper 8 channels (D8-D15) are ignored and only its lower 8 channels (D0-D7) are acquired. These lower 8 channels are sampled subject to different hold clocks on Path B.

00842-025

■ **Figure 4-6. Multiplexed/non-multiplexed input circuitry on the state memory boards.**

4-11

The two hold clocks available for demultiplexing are labeled H1 and H2 on the clock probe. Figure 4-7 illustrates sample and hold clocking and demultiplexing.

In this example, sample clocks S1, S2, and S3 are connected to the $\overline{RD}$, $\overline{WR}$, and $\overline{INTA}$ strobes of the microprocessor. Their polarities are all set to positive (in the Format Menu), because their trailing edges are positive-going. S1 and S5 are not connected, and so are set to off.

The least significant 8 bits of the data probe (bits D0-D7) have been connected to the AD0-AD7 lines on the microprocessor, which carry multiplexed address and data information.
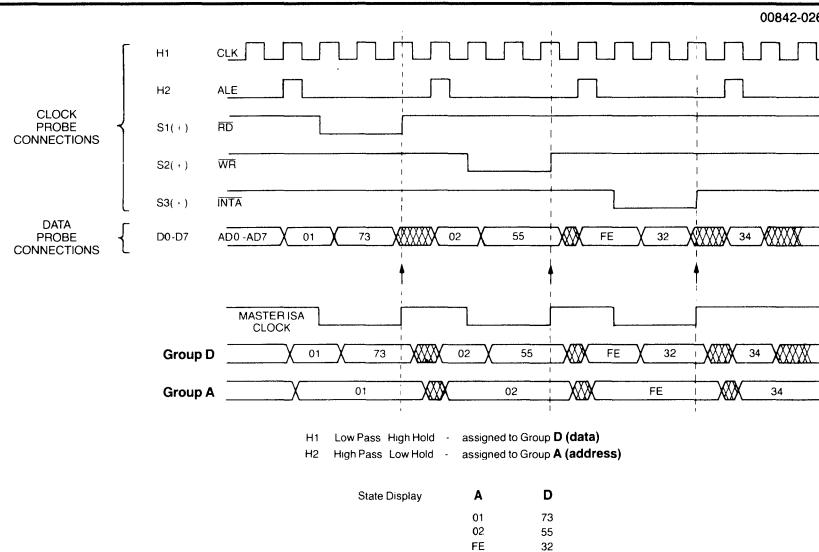
The hold clocks enable the state analyzer to demultiplex this bus and acquire both address and data simultaneously. Inside the analyzer, the probe lines D0—D7 are split into two paths, one for addresses and one for data. Each path is sampled by a hold clock, and the resulting address and data are held until the master clock occurs. When the master sample occurs, the address and data are both stored in memory.

In Figure 4-7, H2 (hold clock 2) is connected to the microprocessor's ALE (Address Latch Enable). H2 has been programmed in the Format Menu to be high pass, low hold. This means that the address and data lines will pass transparently through the H2 presample latch while ALE is high, and will be latched, or held, when ALE goes low. The data gathered by the presample latch is represented in Figure 4-7 by Group A. As you can see from the diagram, the microprocessor's valid address (the lower 8 bits) has been latched and held so it can be clocked into the analyzer by the master clock.

Notice, in Figure 4-7, that the data portion of the address and data (73, 44, 32) has shifted slightly to the right in Group D. This happens because H1 (hold clock 1) is connected to the microprocessor's CLK signal. H1 has been programmed to be low pass, high hold. The microprocessor data is held in the H1 presample latch for nearly half a microprocessor clock period after it has changed on the address/data lines. H1 will generate data hold times only if the microprocessor system allows the data to go invalid before the end of its data strobes ($\overline{RD}$, $\overline{WR}$, and $\overline{INTA}$, for example).

4-12

00842-026

■ **Figure 4-7. Using sample and hold clocks with a multiplexed microprocessor.**

Run/Stop Line

The Run/Stop line is an output generated by the state analyzer to indicate the run/stop status of the analyzer hardware. Its output polarity can be programmed in the Format Menu as either high for run and low for stop, or vice versa.

Real-Time Programmable Outputs

The Real-Time Programmable Outputs, labeled RPO1 and RPO2 on the clock probe, are user-programmable outputs. They can be programmed to change value with the transitions from state to state while the analyzer is running. The RPOs are programmed in the Trigger-Store Menu, where you define what logic levels the RPOs should output at each state transition.

████ TRIGGERING OPTIONS

The flexibility of the Interactive State Analyzer, which provides you with the tools for homing in on the sources of problems, is based on its programmable state machine. Here are its key features:

- Fifteen independent states controlling triggering and state sequencing, storage qualification, and three programmable external outputs (the RPOs) and the Crosslink line (the CL).

- Two word recognizers per state to control triggering, state sequencing, and the programmable outputs.

- Two word recognizers per state for data storage qualification.

You can program the fifteen states to detect sequences of events, including branching sequences. Six variations of IF-THEN-ELSE programming logic are available, as shown in Figure 4-8.

A. IF <event> OCCURS (1-4096) TIMES THEN (GOTO <state> or TRIGGER).

B. IF <event> OCCURS (1-4096) TIMES THEN (GOTO <state> or TRIGGER).
ELSEIF <event> OCCURS ANYTIME THEN (GOTO <state> or TRIGGER).

C. IF <event> OCCURS AFTER (1-4096) (SAMPLE or STORE) CLOCKS THEN
(GOTO <state> or TRIGGER).

D. IF <event> OCCURS AFTER (1-4096) (SAMPLE or STORE) CLOCKS THEN
(GOTO <state> or TRIGGER).
ELSEIF <event> OCCURS ANYTIME THEN (GOTO <state> or TRIGGER).

E. IF <event> OCCURS BEFORE (1-4096) (SAMPLE or STORE) CLOCKS THEN
(GOTO <state> or TRIGGER).
ELSE (GOTO <state> or TRIGGER).

F. IF <event> OCCURS BEFORE (1-4096) (SAMPLE or STORE) CLOCKS THEN
(GOTO <state> or TRIGGER).
ELSEIF <event> OCCURS ANYTIME THEN (GOTO <state> or TRIGGER).

■ **Figure 4-8. IF-THEN-ELSE Triggering Options.**

The simplest trigger mode is mode A:

IF <event> OCCURS (1-4096) TIMES THEN (GOTO <state> or TRIGGER).

In this mode, events are counted until the total matches the preset value, and then the action is taken. Like all versions of the state program, until the IF or ELSEIF condition occurs, data is stored according to the storage definition specified below the two IF-THEN-ELSE definitions in each state. When the definition is satisfied, control is transferred to another specified state or the system is triggered (which means that acquisition ends, subject to the trigger position programmed at the top of the Trigger-Store Menu). During this transition the Real-Time Programmable Outputs (RPOs) and the Crosslink (CL) are set to the new values defined in the state you are leaving.

Trigger mode A is useful for identifying particular events or setting up looping conditions. For example, you could define the state as follows:

IF RESET OCCURS 0001 TIMES THEN TRIGGER.

Data will be qualified and stored continuously until the RESET is detected, and then acquisition will stop. Qualified data leading up to the RESET will be stored and displayed, so you can see what caused this condition.

An additional ORed event added in mode B expands this flexibility. The first event takes precedence over the second if the two occur at the same sample.

Building on the preceding example, a more elaborate definition is:

**IF RESET OCCURS 0001 TIMES THEN TRIGGER
ELSEIF PROC1 OCCURS ANYTIME THEN GOTO <2>,**

where PROC1 stands for procedure 1 (so named in the Symbol Menu).

Now we can look for occurrences of RESET that come before PROC1. If the GOTO <2> were changed to TRIGGER, we would be looking only for RESET or PROC1 to occur before triggering.

Another valuable capability offered in mode C is counting clocks. You count sample clocks (the five system clocks ORed together to generate the master clock, according to specifications in the Format Menu) or store clocks (the sample clocks that correspond to storing qualified data).

The addition of ELSEIF to mode C gives you mode D. Here's a trick for setting up three ORed events:
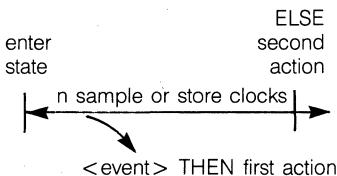
> IF ANY VALUE OCCURS AFTER 0001 STORE CLOCKS
>   THEN GOTO <2>
> ELSEIF <event 3> THEN GOTO <2>
> IF <event 1> OR <event 2> THEN STORE

Event 1 and event 2 cause their respective data to be stored. When one store clock occurs, the top IF line is true. The ELSEIF line is the third ORed event. The disadvantage of this technique is that, for this state, only event 1 or event 2 data is stored. This mode is useful for the ORed recognition of three events which move to another state with less restrictive data qualification.
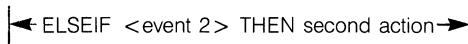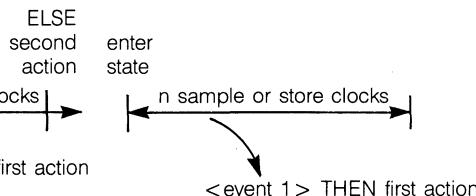
Modes E and F, using the BEFORE parameter, open a window of a specified number of sample or store clocks. If the event occurs when the window is open, the first action will be taken. In mode E, if the event fails to occur before the count is completed, the second action is taken automatically. In mode F, the second action is an ORed condition that can occur at any time before or after the count is completed.

The following diagram illustrates the differences between modes E and F.

Mode E   **Mode F**

                        ELSE
enter   second   enter
state   action   state

|←——— n sample or store clocks ———|→ |←——— n sample or store clocks ———|

        <event> THEN first action

                                                    <event 1> THEN first action

                                    |←— ELSEIF <event 2> THEN second action —→

Let's see how you can define a state or sequence of states that tests indefinitely until a condition is met and the proper data stored. Consider this state definition:

> IF IOERR OCCURS BEFORE 0020 STORE CLOCKS THEN
>   TRIGGER
> ELSEIF TRAP1 OCCURS ANYTIME THEN GOTO <next>,

where IOERR is an I/O error and TRAP1 is a procedure designed to catch the error.

This allows you to narrow in on a very specific problem. You'll find out what causes the I/O error to occur within 20 qualified data events of entering the state. The ELSEIF condition provides a way to restart acquisition when the error has been handled properly.

Several additional examples of state machine programming are described in detail in the demonstration exercises in Section 7.

■ STORAGE QUALIFIER OPTIONS

Figure 4-9 illustrates the storage qualifier options available in the state machine.

A. ALWAYS—> STORE

B. IF / name OCCURS THEN—> STORE

C. IF / name OR / name OCCUR THEN—> STORE

D. IF / name AND / name OCCUR THEN—> STORE

E. NEVER—> STORE

■ **Figure 4-9. Storage qualifier options.**

Two word recognizers are available for data storage qualification in each state. The five modes listed in Figure 4-9 show the combinations available.

The simplest is mode A: always store all sampled data.

Mode E sets the state to never store data, just as mode A sets the state to always store. Mode E is useful in conjunction with the acquisition mode to store all state transitions (as specified in the Trigger-Store Environment Submenu). Then the only storage is the movement between states, that is, the causes of state transitions.

Mode B uses one event to qualify storage. The event can be as loosely defined as a control line, such as store all I/O accesses or all READ accesses, or it can be as tightly specified as a single data value stored into a variable. Similarly, the NOT selection for the event can also be loosely defined, for example, as store all bus cycles except I/O cycles, or tightly defined, for example, as store when the variable is written to a value that is not a given value.

Mode C, the ORing of two events for storage, is useful for more complex storage qualification. In MPA mode, you can see code and stack information leading up to two separate procedures defined by the two events. The ORed combination of two events can also be used to obtain a range of addresses outside the strict binary range with just one event.

For example, suppose you want to trace code accessing a data array that starts at hexadecimal address 1240 and ends at 1284. The address range can be split out as follows:

1240-127F, or 1280-1284

Event 1 is programmed to the value 1240 with a mask FFC0. Event 2 is 1280 with a mask FFFC.

With MPA mode on, this storage qualification will display code that accesses any element of the array. This ORing technique does not support full-range recognition, but it does permit an approximate area of values that can cover the range required, although it may overlap additional values.

Mode D, the ANDing of two events, may seem not to be useful, because all channels are ANDed together to form one count. But it is useful when one event recognizes one portion of the channels and the other is set to a not value of another portion.

For example, to store all writes to address B7F2 with a data value not equal to 20, you would program event 1 for address=B7F2 and event 2 for data=20. The following store line would provide this setup:

IF <event 1> and / <event 2> THEN —>   STORE

■■■■ RUN (GO) MODES

Several run modes are provided by the Interactive State Analyzer. These modes are selected in the Display Environment Submenu, and they are started by pressing G for Go from the command line. Each run mode is introduced below:

- GO ONCE—The analyzer gathers data until a trigger is found. The acquisition continues until memory is filled with the specified amount of post-trigger data. The amount of pre- and post-trigger is specified in the Trigger Position field in the Trigger-Store Menu.

- GO FOREVER—The analyzer takes successive acquisitions. This mode is often used while viewing either the Display or Histogram Menu. The analyzer updates the display or histogram automatically with each new acquisition.

✴ ■ GO TIL ACQ = REF—After each acquisition, the analyzer compares acquisition data to the contents of reference memory. The acquisitions continue until the two memories match. (The comparison parameters are set up in the Display Environment Submenu.)

✴ ■ GO TIL ACQ < > REF—After each acquisition, the analyzer compares acquisition data to the contents of reference memory. The acquisitions continue until a difference between the memories is found.

Both GO TIL ACQ = REF and GO TIL ACQ < > REF can be used to help isolate intermittent problems.

In addition to the above modes, the state analyzer provides a special feature that lets you acquire data until a memory overflow is detected. This feature is useful when you cannot establish an appropriate trigger. Once started, the analyzer periodically checks acquisition memory and stops acquisition the first time it detects a memory overflow condition.

■■■■ STATE DISPLAY & ANALYSIS

Once data is acquired, you can use the Display Menu to view the data in a columnar, radix format. You can also make acquisition and reference memory comparisons.

State Display

The organization of the Display Menu is set up through the Format Menu. You can organize incoming channels into logical groups, with each group containing from one to 32 channels. You can also name the groups and specify their display radices as hexadecimal, octal, binary, or ASCII.

The Display Menu shows the data in columns that reflect the selected channel groups and their radices. Special command-line functions are provided that allow you to scroll the display window to the portion of memory you are interested in. You can page backwards and forwards through memory, or you can jump to any specific memory location.

Reference Memory Comparisons

At any time, you can transfer acquisition data to the reference memory, then compare new acquisitions with old. The boundaries of this data transfer and comparison are set up through the Display Environment Submenu.

Display modes are available that let you view acquisition and reference memories separately, or in comparison. When viewed in comparison, the two memories are displayed side by side and any differences between them are highlighted.

■■■ HISTOGRAM OVERVIEW DISPLAYS

A special feature of the state analyzer is its Histogram Menu.

In this menu, you can perform range-activity measurements of acquired data.

The Histogram Menu works in the following way. You define up to 16 address or data ranges for a specific channel group. The state analyzer then calculates and graphically displays the percentage of acquired data that falls into each range.

Additional functions are provided that allow you make continuous acquisitions while in this menu. This way, you can dynamically monitor the system under test and observe any shifts in activity levels.

■■■ STATE AND TIMING TOGETHER MODES

If you have both the Interactive State Analyzer and the Model 2200 Interactive Timing Analyzer installed in your μAnalyst, you can use them together in time-aligned acquisition modes. The μAnalyst provides a full range of cross-arming and cross-triggering capabilities.

Refer to Section 6, *State and Timing Together*, for a detailed discussion of how to use these two analyzers together.

Using this section. This section details all facets of the menu structures used in the Interactive State Analyzer. It covers the uses, mechanics, and applications of each menu, both in total and in their component parts.

It is recommended that first-time users read the entire section, then later use it as a reference guide.

■■■ SECTION CONTENTS

■■■ MENU STYLE

In Northwest Instrument Systems' products the word *menu* takes on extended meaning. The following paragraphs introduce you to our way of implementing menus.

What are Menus?

Traditionally, *menu* has referred to a simple, one-dimensional list of function or command options. This is also true for the Interactive State Analyzer, but menu also refers to a more visual, two-dimensional method of displaying selectable or programmable items. Fields within a menu may have multiple choices associated with them.

The command line is a list of directives or operations that appears at the bottom of the screen. Typically, these commands move the system from one menu to another, or initiate specific actions within the system or the menu currently displayed.

When you are in the edit mode, the edit prompt line appears in place of the command line. This line indicates the acceptable edit choices for a specific field. The following subsections will have more details about the command and edit modes.

Command and Edit Modes

In the following pages and in using the state analyzer itself, you will notice several command options. One important item to remember is that there are two distinct menu modes:

- **Edit**—to set up or change a programmable field within a menu, you must be in the edit mode.

- **Command**—to run the analyzer as programmed, you must be in the command mode. Other functions of this mode include viewing collected data, using various tools to analyze the data, storing the collected data, and so on.

Visual and Audio Cues

The state analyzer keeps you informed of what it is doing through several reserved lines at the bottom of the screen.

At the bottom of the screen is the command line. This line shows what commands are currently available. Mnemonic key assignments are used wherever possible in this line. For example, you would press E to enter the edit mode, G to go, M to list the menus available, and so on. Keep an eye on this line; the scope of available commands can change from mode to mode and menu to menu.
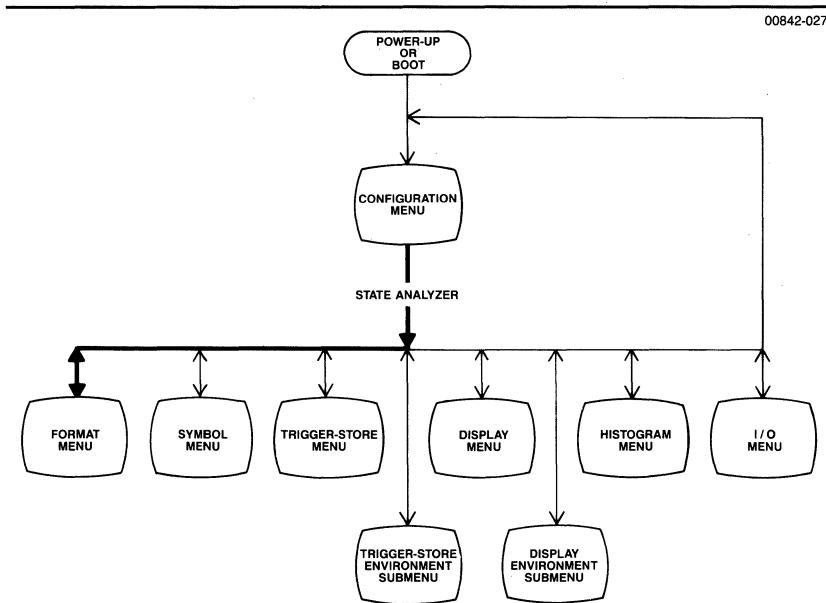
The next line up from the bottom displays errors, warnings, and other messages. Appearance of text on this line is accompanied by some audible signal. A short clicking sound indicates that additional information has been displayed. A beep indicates a warning, a potential error condition, or an error condition. When you hear a click or beep, refer to this line.

Menu Flow and Hierarchy

Figure 5-1 shows the basic structure of the state analyzer menus.

Note that once you are in the Format Menu you can freely jump from menu to menu. (The bold line to the Format Menu indicates the first-time path taken from the Configuration Menu.)

You enter a menu by pressing M for Menus from the command, then by pressing the first letter of the menu name. For example, T for Trigger-Store. You enter an environment submenu by pressing V for enVironment from the command line, then by pressing the first letter of the submenu name.

00842-027

■ **Figure 5-1. Menu Flowchart.**

Programming Fields Within Menus

As you will see later on, there are a lot of things that you can program in any menu to make the analyzer perform as needed. Some basic conventions have been followed throughout the state analyzer system.

1. To change a programmable field, you must be in the edit mode. You can only enter the edit mode when the Edit command appears on the command line at the bottom of the screen.

2. Pressing the escape key exits the edit mode and returns you to the command mode.

3. Once you are in the edit mode, fields that appear in reverse video are programmable. The field that can be modified will blink.

4. Many of the fields are programmed by making selections from a predetermined list of options. When prompted by the edit prompt line "SPACE scrolls through choices," all you need to do is press the space bar to view and select the options.

   When the desired option appears, you can move to another field by using the cursor keys, the return key, or the tab key. Or, you can leave the edit mode by pressing escape.

5. In command or edit mode, your options are listed on the bottom line of the screen. When in doubt about what do next, consult this line. <SP> at the end of the line means that the list of command choices is too long to fit on one line. Press the space bar to see the remaining choices.

Conventions of Use

In edit mode, there are several ways to move to the desired field:

Four Cursor (arrow) Keys — These keys move the screen cursor to the nearest programmable field in the direction selected.

If the field is a text field, the left and right cursor keys move the screen cursor through each individual character position. At either end of a text field, an additional left or right cursor key press moves the screen cursor to the nearest field.

The up and down cursor keys move the screen cursor to nearest field above or below the current field.

Return or Tab Key	Either key moves the screen cursor to the start of the next field. If the screen cursor is in the last field in the menu, either key will move the cursor to the first field in the menu.
Control-Return or Shift-Tab Key	Either key moves the screen cursor to the start of the previous field.
Escape Key	This key exits the edit mode and returns you to the command mode. If you have made an input error prior to pressing the escape key, the system will not exit the edit mode. Instead, it will advise you of the problem and prompt for re-entry of the information. Press the escape key again when the error condition has been corrected.

Command Line Summary

In the command line, some groups of commands are subordinate to others and will only appear after the command has been selected. (For example, the list of menu choices are subordinate to the Menus command.)

The commands described in the following paragraphs are the ones you will see and use most frequently. Additional commands that are specific to a menu are covered in the appropriate portions of this section, and are not described here.

Edit	Press E to enter the edit mode. Edit mode must be entered before any programmable fields on the current screen can be accessed and altered.
	Once you are in edit mode, the edit prompt line appears. This prompt line tells you what edit options are available for a particular field.
	Some edit commands are specific to a particular menu or field within a menu. These edit commands are covered in the appropriate portions of this section.
	Press the escape key to exit the edit mode and return to the command mode.
en**V**ir	Press V to call up the Environment Submenu selections. The command line will list the submenu choices available. Refer to the appropriate portions of this section for details on each submenu listed.

Go   Press G to start the instruments in the μAnalyst. The go mode for the analyzer is determined by the Display Environment Submenu. Refer to that submenu later in this section for a complete listing.

Menus   Press M to call up the menu selections. The command line will list the menu choices available. Refer to the appropriate portions of this section for details on each menu listed.

☑ **NOTE**

The User Menu and User Program Menu selections are not covered in this manual. These menus are only available if you have designed custom software for the Interactive State Analyzer.

Display   Press D to enter the state analyzer's Display Menu. For details of this menu, refer to the appropriate portion of this section.

User__display   Press U to enter a display you have programmed to work with the Interactive State Analyzer. (This type of custom programming is not covered in this manual.)

?=help   Press the ? (question mark) key to call up the μAnalyst's Help Information System. Once you are in the Help system, use the command line choices to move through the help text.

☑ **NOTE**

The Help Information System is only available if you installed the μAnalyst's HELP DISK (refer to Section 3 of this manual).

<SP>   Press the space bar to scroll through the command line choices. This option appears only if the list of choices is too long to fit on the screen.

▮▮▮▮ OVERVIEW
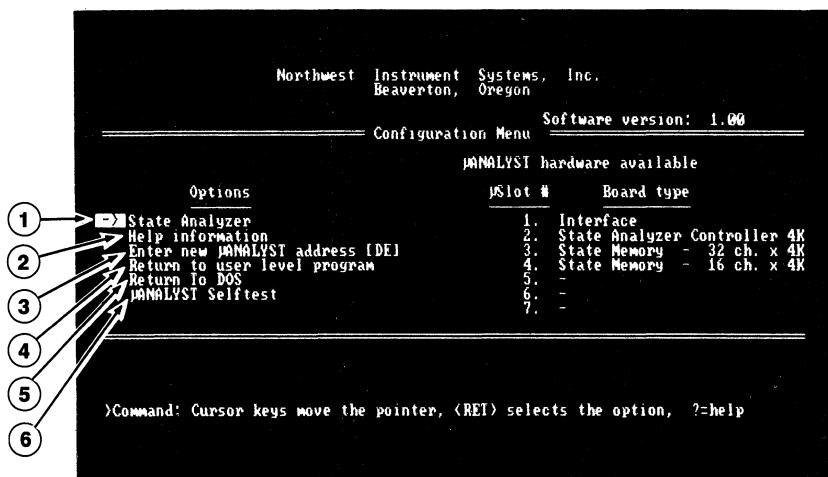
The Configuration Menu is the first menu displayed by the μAnalyst. It tells you the status of the μAnalyst hardware and software, plus it gives you access to the μAnalyst's various operating options.

The operating options available in the Configuration Menu are dependent on the type of hardware you have installed in the μAnalyst mainframe. This section describes the options available for operating the Interactive State Analyzer only. Section 6 of this manual describes the options available for operating the state analyzer in conjunction with the Model 2200 Interactive Timing Analyzer.

▮▮▮▮ OPTION SELECTIONS

Figure 5-2 shows the Configuration Menu and its operating options. Use the numbered callouts in this figure as a reference while reading the following paragraphs. To select an option, simply move the screen pointer to that option, then press the return key.

00842-028

■ **Figure 5-2. The Configuration Menu and its selectable options.**

① State Analyzer

Select this option to enter the menus that control the Interactive State Analyzer. If the system has just been started up, the μAnalyst will enter the state analyzer's Format Menu; otherwise, it will enter the last state analyzer menu used.

You'll find detailed information about how to use the state analyzer menus later in this section.

② Help Information

Select this option to enter the μAnalyst's Help Information System. The μAnalyst will display general information about the state analyzer, its menu conventions, and its keyboard mechanics.

In addition to this option, you can use the ? = Help command to obtain specific information about each option selection. Simply move the screen pointer to the option under question, then press the ? (question mark) key.

③ Enter New μAnalyst Address

Select this option only if you need to enter a new memory range for the PC interface board (see Section 2's *Installing the PC Interface Board* subsection).

☞ **NOTE**

This option is only available on system start-up. It will not be available once any of the state analyzer menus have been viewed or used.

The default setting for this option is DE hexadecimal, which matches the default setting of the PC interface board. If you have changed the setting on the interface board, enter the corresponding 2-digit code, in hexadecimal notation, into this option's field.

If the μAnalyst SYSTEM DISK is not write-protected, the μAnalyst stores the new address value. This way, whenever you restart the system, the new address is automatically entered.

④ Return to User Level Program

If custom software is installed, you can use this option to return to your program.

(5) **Return to DOS**

Select this option to return to the DOS operating software. The μAnalyst will prompt you with a message reading:

"Have you saved your SETUP and DATA? Do you really want to return to DOS (y/N/?)"

This message is a reminder that once you leave the Configuration Menu to return to DOS, all menu setups and memories are lost.

If you have already saved your setups via the state analyzer's I/O Menu, or if you don't want to save your setups, press Y for Yes. The μAnalyst will exit to DOS.

If you want to save your menu setups and have not yet done so, press N for No. Then enter the state analyzer's I/O Menu and save your setups. When you're finished, return to the Configuration Menu and exit to DOS.

☑ **NOTE**

Procedures for using the I/O Menu are provided later in this section.

(6) **μAnalyst Selftest**

Select this option if you want to run the μAnalyst's internal diagnostics. These diagnostics perform a first-fault, static test of the μAnalyst boards. They take several minutes to execute.
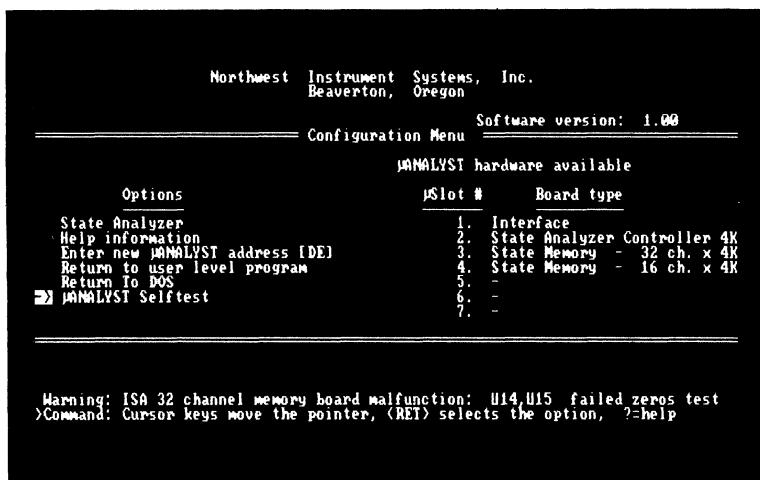
While the diagnostics are being executed, the μAnalyst updates the display screen to tell you which board is being tested and the results. If no faults are found in the board being tested, the μAnalyst will tell you that the board has passed the selftest. If a fault is found, the μAnalyst will display a coded message similar to that shown in Figure 5-3.

If a board fails the selftest, contact Northwest Instrument Systems, Inc.

00842-029

```
                    Northwest Instrument Systems, Inc.
                    Beaverton, Oregon

                                          Software version: 1.00
                        ======= Configuration Menu =======

                                  µANALYST hardware available

          Options                    µSlot #     Board type

    State Analyzer                    1.  Interface
    Help information                  2.  State Analyzer Controller 4K
    Enter new µANALYST address [DE]   3.  State Memory  -  32 ch. x 4K
    Return to user level program      4.  State Memory  -  16 ch. x 4K
    Return To DOS                     5.  -
 ➠) µANALYST Selftest                 6.  -
                                      7.  -


  Warning: ISA 32 channel memory board malfunction: U14,U15 failed zeros test
 )Command: Cursor keys move the pointer, <RET> selects the option,  ?=help
```

■ **Figure 5-3. An example of a selftest error.**

███ OVERVIEW

The Format Menu allows you to set up acquisition parameters for the system under test and to format acquired data.

In this menu, you define the state analyzer's flexible clocking scheme. Five sample clocks are ORed to make up the master clock. Two hold clocks allow you to delay, or stretch, data until a valid sample clock samples the data. A combination of sample and hold clocking and the multiplexed data acquisition mode enable you to capture data from multiplexed systems.

For the clock probe, you can set the polarity of the run/stop line. For the data probes, you can specify multiplexed or non-multiplexed data acquisition.
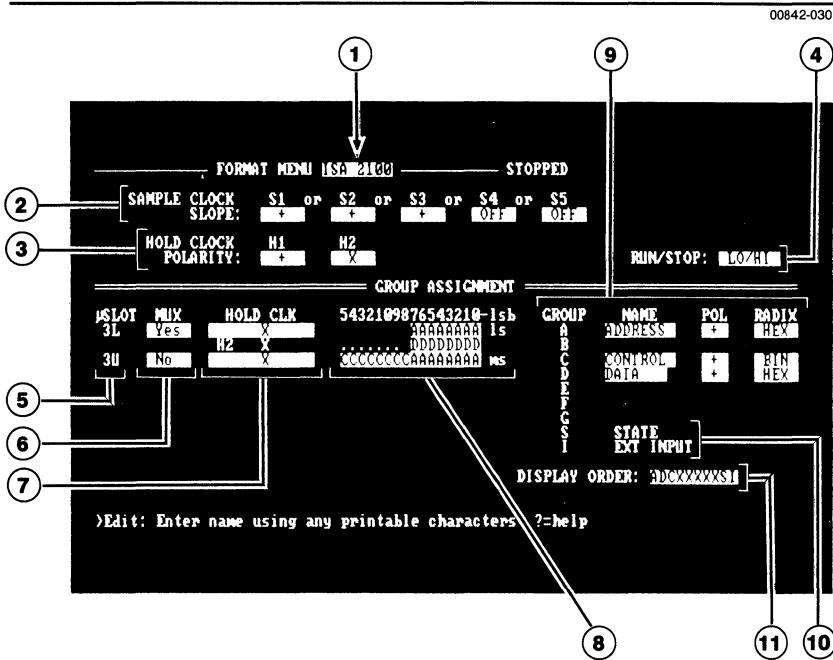
To make the Symbol and Display Menus easier to interpret, you can assign labels to channel groups, specify their appropriate radix and polarity, and predetermine their display order. This subsection explains how you use the Format Menu to perform all of these activities.

███ MENU FIELD SELECTIONS

Figure 5-4 illustrates a typical Format Menu display. You enter this menu by pressing M for Menu, then F for Format from the command line.

Use the numbered callouts in Figure 5-4 as a reference while reading the following field descriptions. To make any changes to a field, you must be in edit mode (press E for Edit), and you must have the screen cursor positioned in the field.

If the HELP DISK is installed, you can use the ?-Help function to obtain information about a specific field. To do this, enter the edit mode, then move the screen cursor to the desired field and press the ? (question mark) key.

00842-030

■ Figure 5-4. The Format Menu and its field selections.

(1) Name

This field lets you name your menu setups. The name you enter here will appear at the top of all other state analyzer menus, and it will be automatically stored with any menu setups or memories in the I/O Menu.

The default name in this field is ISA 2100. You can change this to any other name using up to eight characters. A useful name might be the name of the microprocessor or test procedure you are working with, such as 8086__SYS or MEMCHECK.

(2) Sample Clock Slope

These fields are used to establish the synchronous, master sampling clock. The five sample lines, labeled S1—S5, are provided via the clock probe attached to the state analyzer's controller board. Each line provides a maximum sampling rate of 10 MHz.

You can use one or all of these sample lines to define the master clock. The lines work together in an ORed condition.

Using the space bar, you can set the polarity of any sample line to:

■ + (positive polarity)—data is sampled on the signal's rising edge.

■ — (negative polarity)—data is sampled on the signal's falling edge.

■ OFF—the sample line is not used.
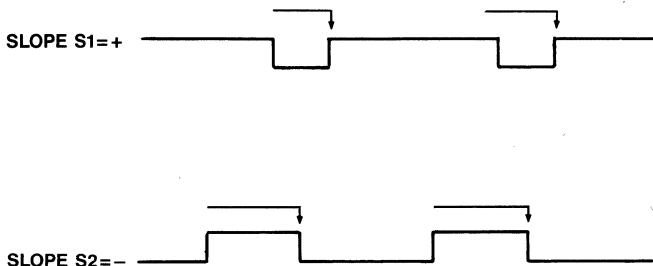
☑ **NOTE**

If you turn a sample line off, make sure that the line is physically disconnected from any signal source. Otherwise, it may cause interference with the other sample lines.

Positive vs Negative Clock Polarity. Figure 5-5 illustrates the difference between positive and negative clock polarity.

When positive polarity is selected for a clock, the data is sampled on the clock's rising edge, and the pulse width for the clock is defined from falling edge to rising edge. When negative polarity is selected for a clock, the data is sampled on the clock's falling edge, and the pulse width for the clock is defined from rising edge to falling edge.
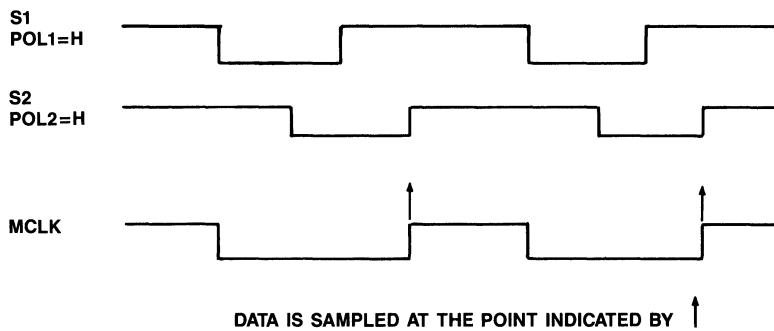
■ **Figure 5-5. Sample clock polarity.**

Overlapping Clocks. The five sample clocks are ORed to produce the master clock. When the sample clocks do not overlap in time, they produce a master clock consisting of pulses contributed by each of the sample clock pulses. When the sample clocks do overlap, the master clock combines the pulses to create a longer pulse.

An example of overlapping sample clocks is provided in Figure 5-6. Data sampling occurs on the rising edge of the master clock (MCLK).

00842-032

DATA IS SAMPLED AT THE POINT INDICATED BY ↑

■ **Figure 5-6. Overlapping sample clocks.**

③ **Hold Clock Polarity**

These fields are used to set the polarity of the two external hold clocks. The hold clocks, labeled H1 and H2, are provided by the clock probe. Using the space bar, you can set the polarity of either hold clock to:

■ + (positive polarity)

■ − (negative polarity)

■ X (off)

Depending on how the hold clocks are being used, the selected polarity may refer to level or edge clocking. (For details, see *Hold Clock* field, callout number 7.)

④ Run/Stop

This field lets you program the polarity of the Run/Stop line. The line is an open collector output from the clock probe, and it is used to indicate to external hardware when the analyzer is running (acquiring data) and when it is stopped.

Using the space bar, you can set the polarity of the Run/Stop line to:

■ HI/LO—the line is high when the analyzer is running and low when it is stopped.

■ LO/HI—the line is low when the analyzer is running and high when it is stopped.

⑤ µSlot (Probe Identifiers)

This menu column is not a programmable field. Rather, it serves as a probe identifier for the MUX, Hold Clock, and Channel/Group Assignment fields (see callout numbers 6, 7, and 8).

In this column, each probe is identified by its slot number. An *L* indicates the lower probe on a 32-Channel Memory Board, while a *U* indicates the upper probe. A 16-Channel Memory Board has only one probe, so there is not an *L* or *U* designation.

⑥ MUX (Multiplex)

These fields allow you to establish multiplexed or non-multiplexed acquisition for each data probe.

Using the space bar, you can set each probe to:
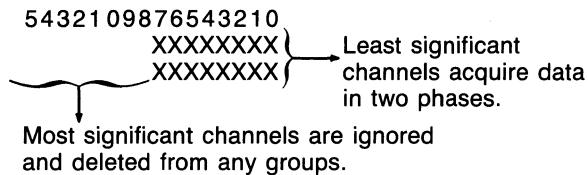
■ YES—the probe acquires multiplexed data.

■ NO —the probe does not acquire multiplexed data.

When you first attempt to set a probe's MUX field to YES, the µAnalyst will prompt you with the following message:

"Change channel assignments to keep groups contiguous? (y/N/?)"

This message is a reminder that when a probe is set to multiplexed acquisition, its eight most significant channels are ignored and removed from any channel groups. The multiplexed data is acquired in two phases on the eight least significant channels.

Verify that you want to select multiplexed acquisition by pressing Y for yes. The probe channels will be automatically rearranged as shown below.

```
5432109876543210
        XXXXXXXX ⎫      Least significant
        XXXXXXXX ⎭      channels acquire data
                        in two phases.
```

Most significant channels are ignored
and deleted from any groups.

☞ **NOTE**

Multiplexed Clocking and channel-to-group assignments are discussed under the next two menu fields, *Hold Clock* and *Channel/Group Assignments,* callout numbers 7 and 8.

⑦ **Hold Clock**

These fields are used to assign the hold clocks, H1 or H2, to specific data probes.

The characteristics of the two hold clocks are determined by the polarities set up in the Hold Clock Polarity fields (see callout number 3). Table 5-1 describes the various hold-clock selections, and explains when they are available for use.

When acquiring non-multiplexed data, you can program the probe's Hold Clock field to any of the available selections listed in Table 5-1.

When acquiring multiplexed data, you can program the Hold Clock field for the first data byte to any available selection, but you cannot program the Hold Clock field for the second data byte. As shown below, the second data byte's Hold Clock field is always set to H2 (either HI = PASS or LO = PASS, depending the H2 polarity).

	HOLD CLK	5432109876543210
Byte 1 (programmable)	X	XXXXXXXX
Byte 2 (non-programmable)	H2	XXXXXXXX

A detailed example of how to use the hold clocks is provided in Section 7 of this manual.

■ **Table 5-1: Hold Clock Selections**

Field Selection	Description
X	This selection is always available. It turns the hold clock field off. The probe will acquire data using the master sampling clock only.
H1 X (H2 X)	This selection is available when the H1(H2) hold clock is turned off (X) in the Hold Clock Polarity field. It has no affect on acquisition. The probe will acquire data using the master sampling clock.
H1 ∧ EDGE (H2 ∧ EDGE)	This selection is available only if the H1(H2) hold clock is set to a positive (+) polarity, and if you are using a 32-channel memory board. (16-channel memory boards only allow level clocking).
	With this selection, the probe will latch data on the rising edge of the hold clock, then hold it until a master sampling clock occurs.
H1 ∨ EDGE (H2 ∨ EDGE)	This selection is available only if the H1(H2) hold clock is set a negative (-) polarity, and if you are using a 32-channel memory board. (16-channel memory boards only allow level clocking).
	With this selection, the probe will latch data on the falling edge of the hold clock, then hold it until a master sampling clock occurs.
H1 HI=PASS (H2 HI=PASS)	This selection is available when the H1(H2) hold clock is set to a positive (+) polarity.
	With this selection, the probe will pass data to its latches while the hold clock's level is high, then hold that data while the hold clock's level is low. The data will then be clocked into memory when a master sampling clock occurs.
H1 LO=PASS (H2 LO=PASS)	This selection is available when the H1(H2) hold clock is set to a negative (−) polarity. With this selection, the probe will pass data to its latches while the hold clock's level is low, then hold that data while the hold clock's level is high. The data will then be clocked into memory when a master sampling clock occurs.

⑧ **Channel/Group Assignments**

These fields let you organize data channels into meaningful display groups. Seven groups, labeled A—G, are available for use. Each group can contain up to 32 channels.

To assign a channel to a group, simply enter the group letter into the channel's field. To set a channel to don't care, enter an X.

All channels within a given group must be contiguous; that is, they must be linked from least significant bit (lsb) to most significant bit (msb). The following paragraphs explain how to determine the bit significance for both non-multiplexed and multiplexed channels.

Bit Significance of Non-Multiplexed Channels. With non-multiplexed channels, the least significant bit (lsb) is the lowest numbered channel of the lowest numbered probe. The most significant bit (msb) is the highest numbered channel of the highest numbered probe.

For example, if your display shows three probes, the bit significance of their channel fields would read as:

```
                          5432109876543210
(lowest numbered probe)   XXXXXXXXXXXXXXXX ◄── lsb
                          XXXXXXXXXXXXXXXX
              msb ──► XXXXXXXXXXXXXXXX
```

Bit Significance of Multiplexed Channels. The bit significance of the multiplexed channels is linked by first and second byte acquisitions. The first bytes (the top channel fields) of all multiplexed probes are linked together, and so are the second bytes (the second channel fields).

For example, suppose you were using two probes to demultiplex a 16-bit address/data bus. The channels of the two probes would be linked as shown below:

```
                      5432109876543210
Probe    Byte 1       AAAAAAAA      lsb
         Byte 2       DDDDDDDD              lsb
Probe                                │      │
         Byte 1       AAAAAAAA      msb     ▼
         Byte 2       DDDDDDDD              msb
```

⑨ Group Name, Polarity, and Radix

These fields are used to assign names, logic polarities, and radices to your channel groups.

Enter a group name using up to eight characters. Useful names might be ADDRESS, DATA, STATUS, CONTROL, etc.

Select a group's polarity using the space bar. Choices include: positive (+) or negative (−). If positive polarity is selected, the Display Menu will show data above the probe threshold as high (1) and data below the probe threshold as low (0). If negative polarity is selected, the Display Menu will show data above the probe threshold as low (0), and data below the probe threshold as high (1).

☞ **NOTE**

The Trigger-Store Menu automatically adjusts its word recognizers to match the selected polarity.

Select a group's radix by using the space bar. Choices include: HEX (hexadecimal), OCT (octal), BIN (binary), or ASC (ASCII). If ASCII is selected, only the seven least significant bits of the group determine the ASCII character (see Appendix A at the back of this manual).

⑩ STATE and EXT INPUT Groups

These groups are not programmable. They are reserved for status information that is always acquired and available for display. The STATE group is reserved for trigger-store state information, while the EXT INPUT group is reserved for data received via the External Input connector.

⑪ Display Order

This field is used to specify the display order of the channel groups. You can assign any order to the groups using the characters A—F, S, and I. The selected group order is then reflected in the Symbol and Display Menus.

You can also inhibit the display of any group by entering an X for don't care. The inhibited group does not appear on the state display, but it is still available for defining word recognizers in the Symbol Menu.

■■■ OVERVIEW

The Symbol Menu enables you to program values associated with the names you've assigned to channel groups in the Format Menu. Once the values are assigned, the symbols can be used in the Trigger-Store Menu and the Display Environment Submenu as identifiers for word recognizers and find words. This means you can refer to user-created events by name throughout the menu setups, making them easier to read and understand.

When working in the Symbol Menu, you may enter the symbol name, enter the value you wish to assign to the name, insert a line to add a new symbol to an already filled-in table, or delete a line that is in error or is no longer needed.

A symbol table is created in one of two ways. You can create symbols in the Trigger-Store Menu, and then go back and enter their values into the Symbol Menu. (This is the more usual approach, because it's easier.) Or you can first create symbols and values in the Symbol Menu, and then enter the symbols by name in the Trigger-Store Menu as part of the triggering and qualification conditions. You can also buffer data from the Display Menu and use it to assign a value to a symbol.

This subsection shows you how to use the Symbol Menu to enter new symbols or modify existing symbols.

■■■ MENU FIELD SELECTIONS

Figure 5-7 illustrates a typical Symbol Menu display. To enter this menu, press M for Menu, then S for Symbol from the command line.
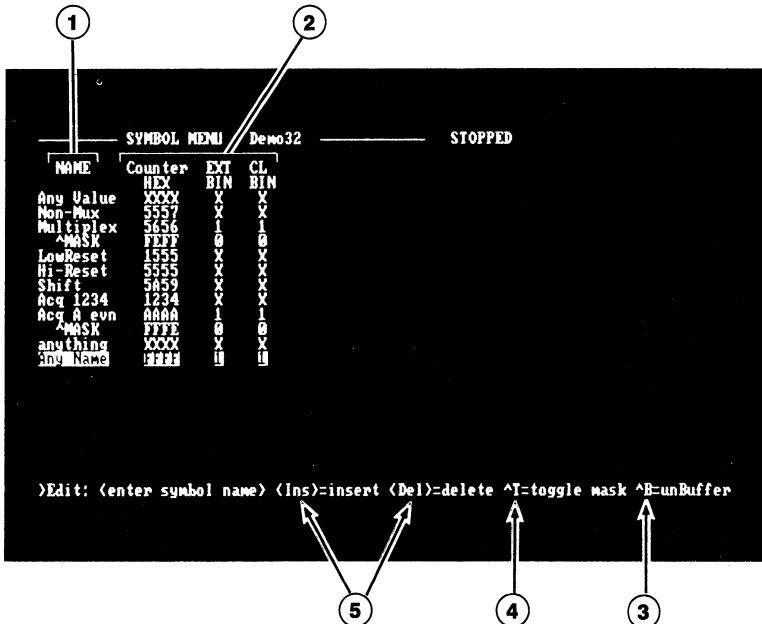
Use the numbered callouts in Figure 5-7 as a reference while reading the following field descriptions. To program a menu field, you must be in the edit mode (press E for Edit), and you must have the screen cursor positioned in the field.

If the μAnalyst's HELP DISK is installed, you can use the ?=Help function to obtain information about a specific field. To do this, enter the edit mode, then move the screen cursor to the desired field and press the ? (question mark) key.

00842-033

■ Figure 5-7. The Symbol Menu and selectable fields.

(1) Name

This field is used for entering, deleting, or modifying symbol names.

A minimum symbol table consists of two symbol names: Any Value and Any Name. These two names are automatically entered into the symbol table at system start-up.

The Any Value name is a permanent symbol that cannot be modified or deleted. It will always reside at the top of the symbol table. Its value is made up of all don't care (X) entries and it can be used to match any event in a trigger-store state. The second default symbol, Any Name, can be modified but not deleted.

To insert a new symbol name, move the screen cursor to an existing symbol name (e.g., Any Name). Then, press the insert key. A new, blank Name field will appear at the cursor location. Enter any desired name up to nine characters.

To modify an existing symbol name, simply enter a new name over the old one.

To delete a symbol name, move the screen cursor to the Name field of the symbol you want to delete. Then, press the delete key. Both the symbol name and its value will be deleted from the table.

☞ **NOTE**

If you try to delete a symbol name that is currently being used by a menu, the μAnalyst will ask you to confirm the operation by pressing Y for Yes. Menus that use symbols include the Trigger-Store and Histogram Menus, and the Display Environment Submenu.

② Data Values

These fields are used for assigning data values to the symbol names.

☞ **NOTE**

An alternate procedure for entering symbol values is discussed under callout number 4, *unBuffer*.

The column headings above the data value fields correspond to the channel groups and radices assigned in the Format Menu. When entering values, you must use the appropriate radix notation, as defined below:

■ HEX—Enter 0 through F, or enter X for don't care.

■ OCT—Enter 0 through 7, or enter X for don't care.

■ BIN—Enter 0 or 1, or enter X for don't care.

■ ASC—Enter an ASCII character or control character (see Appendix A at the back of this manual).

A special condition may arise when you are entering a value. The first digit entry may have fewer legal values than what is usually available under the radix. This occurs when the number of channels assigned to the group is fewer than what is expected by the radix.

For example, a group may have six channels and a hexadecimal radix. This would mean that the first character in the group can only have a value of 0 through 3, or X.

If such a condition should arise, the message line at the bottom of the screen will prompt you accordingly.

③ **unBuffer**

As an alternative to typing in data values, you can use this command to transfer a data value directly from a line in the state display.

To use this command, you must first go to the Display Menu and enter one of the display lines into the buffer (see the *Display Menu* later in this section).

Once you have buffered a line, return to the Symbol Menu. Position the screen cursor in the Name field of the symbol you want to receive the buffered value. Then, press Control-B (∧ B) for unBuffer.

The buffer can only hold one line, so if you wish to make several entries this way, you must return to the Display Menu after one unbuffering operation, place another value in the buffer, go back to the Symbol Menu, and enter the next buffered value.

④ **Toggle Mask**

The Control-T (∧ T) Toggle Mask command lets you set individual masked bits without changing their radix to binary. The command toggles between masked and non-masked values. With masked values, the symbol value and mask are split into two lines, and with non-masked values they are merged.

Each symbol may be thought of as containing two words. The first word is the value of the symbol, and the second word, or mask word, represents which channels are considered as part of the value (cares) and which are not (don't cares). Care bits are represented by 1 and don't care bits by 0.

It is usually convenient to define a symbol as having a specific value in a manageable radix. For example, suppose you were looking for modifications made to hexadecimal addresses in the range D8 to DF. It's much easier to deal with D8 than with its binary representation 11011000. In binary radix, you could define a word to cover the hexadecimal range D8 to DF as 11011XXX. Then in hexadecimal you could define the symbol for the address D8 and assign a mask value F8. The lower three bits are treated as don't cares, but the readability and sense of address D8 are preserved.

For non-masked symbols, enter X into a character position to define the don't care. Depending on the radix of the character position, the don't care may apply to one or more bits. For example, an X entered into a hexadecimal character sets all four bits of the character to don't care.

When merging the symbol value and symbol mask, you may generate a question mark (?) character as part of the value (see Figure 5-8). The question mark indicates that some bits within the boundary of that character (determined by the radix) are set to don't cares.

00842-034

```
----------- SYMBOL MENU   Demo32  -----------   STOPPED

   NAME    Counter  EXT  CL
             HEX    BIN  BIN
Any Value   XXXX     X    X
Non-Mux     5557     X    X
Multiplex   5656     1    1
   ^MASK    FEFF     0    0
LowReset    1555     X    X
Hi-Reset    5555     X    X
Shift       5A59     X    X
Acq 1234    1234     X    X
Acq A evn   AAAA     1    1
   ^MASK    FFFE     0    0
anything    XXXX     X    X
demomask▮   FC18     0    0
   ^MASK    F87F     1    0
Any Name    FFFF     1    1

   )Edit: (enter symbol name) (Ins)=insert (Del)=delete ^T=toggle mask ^B=unBuffer
```

```
----------- SYMBOL MENU   Demo32  -----------   STOPPED

   NAME    Counter  EXT  CL
             HEX    BIN  BIN
Any Value   XXXX     X    X
Non-Mux     5557     X    X
Multiplex   5656     1    1
   ^MASK    FEFF     0    0
LowReset    1555     X    X
Hi-Reset    5555     X    X
Shift       5A59     X    X
Acq 1234    1234     X    X
Acq A evn   AAAA     1    1
   ^MASK    FFFE     0    0
anything    XXXX     X    X
demomask▮   F??8     ?    ?
Any Name    FFFF     1    1

   )Edit: (enter symbol name) (Ins)=insert (Del)=delete ^T=toggle mask ^B=unBuffer
```

■ **Figure 5-8. The Toggle-Mask Option.**

5-27

⑤ Insert and Delete

These two commands are useful when you are entering symbol names. By pressing the insert key, you can insert a new, blank Name field at the screen cursor location. Conversely, by pressing the delete key, you can delete the symbol Name field and its value.

For more information about inserting and deleting symbols, refer back to callout number 1, *Name*.

■■■ OVERVIEW

The Trigger-Store Menu is used to determine which data is stored in memory, and when. You can set up as many as 15 trigger states, with four states displayed on the screen at one time. These states can be used independently, or they can be combined in any sequential or branching order.

Within each state, you can specify one or two word recognizers for defining IF-THEN-ELSE triggering conditions, and you can specify one or two word recognizers for defining storage qualification. You can also assert the Real-Time Programmable Outputs (RPOs) and Crosslink (CL) output for interaction with the system under test.

In the rest of this subsection, you'll find detailed information on how to use the Trigger-Store Menu to set up the various triggering and storage qualifier conditions.

☞ **NOTE**

In portions of this subsection, reference is made to parameters established within the Trigger-Store Environment Submenu. We will discuss these parameters and show how they are used in the next subsection.

■■■ MENU FIELD SELECTIONS

Figure 5-9 illustrates a typical Trigger-Store Menu display. You enter this menu by pressing M for Menu, then T for Trigger-Store from the command line.
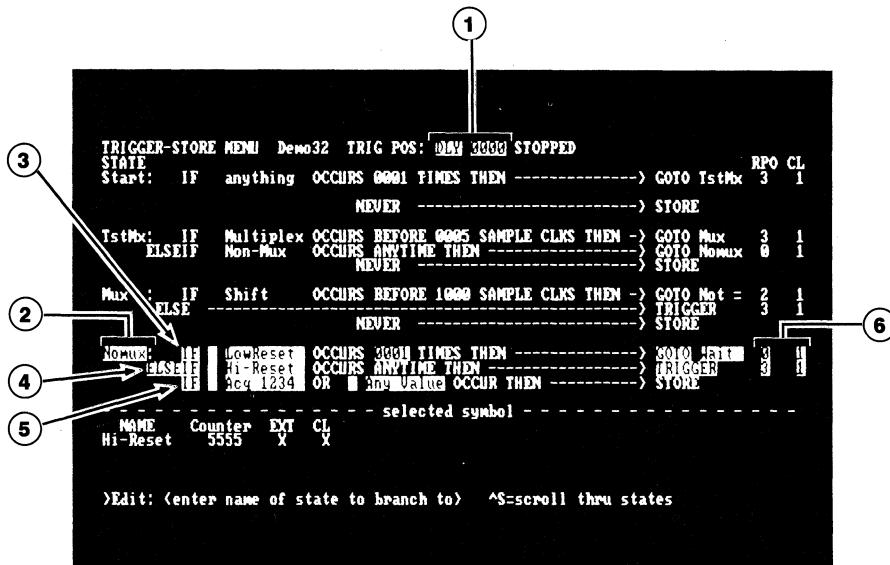
Use the numbered callouts in Figure 5-9 as a reference while reading the following field descriptions. To change a menu field, you must be in the edit mode (press E for Edit), and you must have the screen cursor positioned in the field.

If the μAnalyst's HELP DISK is installed, you can use the ?=Help function to obtain information about a specific field. To do this, enter the edit mode, then position the screen cursor in the desired field and press the ? (question mark) key.

00842-035

Figure 5-9. The Trigger-Store Menu and its selectable fields.

① **Trigger Position**

This field is used to position the trigger in acquisition memory. Using the space bar, you can select:

- BEGIN—the trigger is delayed by 4082 words.

- CENTER—the trigger is delayed by 2047 words.

- END—the trigger is delayed by 13 words.

- DELAY <0—4096>—the trigger is delayed by the specified number of words.

The total acquisition memory depth is 4096 words. Data acquired before the trigger is called pre-trigger data, and data acquired after the trigger is called post-trigger data.

The amount of pre-trigger data acquired in memory depends on how quickly the trigger occurs. The state analyzer looks for a trigger immediately; it does not wait until memory is full.

☞ **NOTE**

When the state analyzer is operating in the MultiplePreview Acquisi-
tion (MPA) mode, the only trigger position available is DELAY
<0—4096>. (Refer to the *Trigger Environment Submenu* contained
later in this section.)

(2) **State**

This field is used to name, insert, or delete trigger states.
Altogether, there are 15 trigger states available for use. Each state
provides two word recognizers for establishing trigger conditions,
and two word recognizers for establishing storage qualification.

To insert a state, move the screen cursor to a current State
field, then press the insert key. The current State field will move
down one position on the screen, and a new, blank State field will
appear at the cursor location.

☞ **NOTE**

You can also create a new state by entering the name of the state
into one of the GOTO fields (see callout number 3).

To name a state, enter up to five characters. To rename an
existing state, simply enter the new name over of the old one. The
μAnalyst will ask you to confirm the renaming by pressing Y for Yes
or N for No.

☞ **NOTE**

When you rename a state, all references to that state (as in GOTO
trigger actions) will be automatically updated.

To delete a state, move the screen cursor to the State field
you want deleted. Then, press the delete key. The μAnalyst will
delete the state and its word recognizers.

☞ **NOTE**

When you delete a state, all references to that state (as in GOTO
trigger actions) will default to the next sequential state. If there are
no next states, the references will default to their own state.

(3) **1st Word Recognizer**

The 1st Word Recognizer can be used by itself, or it can be
combined with the 2nd Word Recognizer to form IF-THEN-ELSE trig-
gering conditions. The possible options for the 1st Word Recognizer
are listed below.

A. IF (/) <symbol> OCCURS (1-4096) TIMES THEN—> (GOTO
   <state> or TRIGGER)

B. IF (/) <symbol> OCCURS AFTER (1-4096) (SAMPLE or STORE)
   CLOCKS THEN —> (GOTO <state> or TRIGGER)

C. IF (/) <symbol> OCCURS BEFORE (1-4096) (SAMPLE or
   STORE) CLOCKS THEN —> (GOTO <state> or TRIGGER)

As you can see, each word-recognizer option has several
programmable fields. The following paragraphs describe how you
make entries into these fields.

Logical True or Not True (/) Field. Use this field to specify
whether the symbol used in the word recognizer is set to its logical
true or not true value. Your choices include:

- ▪ A blank field—the symbol is set to its true value and recogni-
  tion will occur when the symbol value appears at the probe
  tips.

- ▪ / (not)—the symbol is set to its not true value and recognition
  will occur when any value other than the symbol value
  appears at the probe tips.

Symbols. Use this field to enter the symbol you want to use
for word recognition. The value of the symbol is defined in the Sym-
bol Menu.

To use a symbol that has already been defined in the Symbol
Menu, simply enter the symbol's name. If you can't remember all of
the available symbols and their names, you can use the Control-S
($\wedge$ S) key to scroll through the symbols. Each symbol and its value
will appear at the bottom of the screen under the heading "selected
symbol."

You can also enter a symbol that has not been defined in the
Symbol Menu. To do this, simply enter any name using up to eight
characters. The μAnalyst will then ask if you want to add the name to
the Symbol Menu. Press Y for yes.

When you are finished with the Trigger-Store Menu, you will
need to return to the Symbol Menu and enter the values for any new
symbol names you have created.

Occurrence Counter (1—4096). Use this field to set up the
word-recognizer counter. Depending on the word-recognition option
you've selected, the field may serve as an occurrence counter or a
time window. Valid entries into the field include 1—4096, inclusive.

The counter can be set to count SAMPLE or STORE clocks.
Sample clocks are defined by the master sample clock. Store clocks,
on the other hand, are not physical clocks. Rather, they are defined
by the words you have asked the analyzer to store (see *Storage
Qualifiers,* callout number 4).

GOTO <state> or **TRIGGER.** Use this field to set up the word-recognizer action.

When you set a word recognizer to GOTO, you must also enter the name of a state. The name can belong to an existing state, or it can be a new one. If you enter a new name, the μAnalyst will ask if you want to insert a new state. Press Y for yes. Then go to the newly entered state and define it.

☑ NOTE

If you have already defined 15 states, the μAnalyst will not allow you to insert a new state from the GOTO field. Instead, a message will appear reading "no new states are available; must use an existing state name."

④ 2nd Word Recognizer

The 2nd Word Recognizer can be used in conjunction with the 1st Word Recognizer. It allows you to set up the IF-THEN-ELSE triggering conditions outlined below.

A. IF (/) <symbol> OCCURS (1-4096) TIMES THEN —> (GOTO <state> or TRIGGER)

ELSEIF (/) <symbol> OCCURS ANYTIME THEN —> (GOTO <state> or TRIGGER

B. IF (/) <symbol> OCCURS AFTER (1-4096) (SAMPLE or STORE) CLOCKS THEN —> (GOTO <state> or TRIGGER)

ELSEIF (/) <symbol> OCCURS ANYTIME THEN —> (GOTO <state> or TRIGGER

C. IF (/) <symbol> OCCURS BEFORE (1-4096) (SAMPLE or STORE) CLOCKS THEN —> (GOTO <state> or TRIGGER)

ELSEIF (/) <symbol> OCCURS ANYTIME THEN —> (GOTO <state> or TRIGGER

D. IF (/) <symbol> OCCURS BEFORE (1-4096) (SAMPLE or STORE) CLOCKS THEN —> (GOTO <state> or TRIGGER)

ELSE ————————————>(GOTO <state> or TRIGGER

As you can see, the various word-recognizer options have several programmable fields, including: Logical True and Not True (/), Symbols, and GOTO or TRIGGER actions. You program these fields as specified under *1st Word Recognizer*, callout number 3.

⑤ Storage Qualifiers

The third line of each state is used for establishing storage qualifier conditions. These storage qualifier conditions determine which data, if any, the analyzer will store while it is executing the state. The possible options for the storage qualifiers are listed below.

A. ALWAYS ——————————————————————— > STORE

B. IF (/) <symbol> OCCURS THEN ————————————— > STORE

C. IF (/) <symbol> OR (/) <symbol> OCCUR THEN —> STORE

D. IF (/) <symbol> AND (/) <symbol> OCCUR THEN –> STORE

E. NEVER ———————————————————————— > STORE

Options A and B have no additional programmable fields. Options B, C, and D have programmable Logical True and Not True (/) fields and Symbol fields. You program these fields as specified under *1st Word Recognizer*, callout number 3.

⑥ RPO and CL Outputs

These fields let you assert the Real-Time Programmable Outputs (RPOs) and Crosslink (CL) signals.

Real-Time Programmable Outputs (RPOs). The RPOs are two output lines provided via the state analyzer's clock probe. The lines are labeled RPO1 and RP02.

Using the space bar, you can set the RPOs to two bits encoded as 0, 1, 2, or 3. The encoding works as shown in Table 5-2. A zero (0) corresponds to a low-level (open collector, actively pulled down) output and a one (1) corresponds to a high-level (open collector, resistively pulled up) output.

■ **Table 5-2: RPO Output Values**

Field Selection	RPO2	RPO1
0	0	0
1	0	1
2	1	0
3	1	1

The RPO values are output to the clock probe when a transition from the current state to the next state occurs. Thus, while data is being acquired in a particular state, the RPO lines are set according to the value in the previous state.

☑ **NOTE**
The starting values of the RPOs are programmed via the Trigger-Store Environment Submenu.

Crosslink (CL). The Crosslink signal is output through the CL BNC connector located on the back of the μAnalyst mainframe. It is also output internally on the μAnalyst's interconnect bus, and it can be used for crosslink triggering between the state analyzer and other μAnalyst products, such as the Interactive Timing Analyzer.

Using the space bar, you can program the Crosslink to output a 0 (active low) or a 1 (open collector high). The Crosslink value is output when a transition from the current state to the next state occurs.
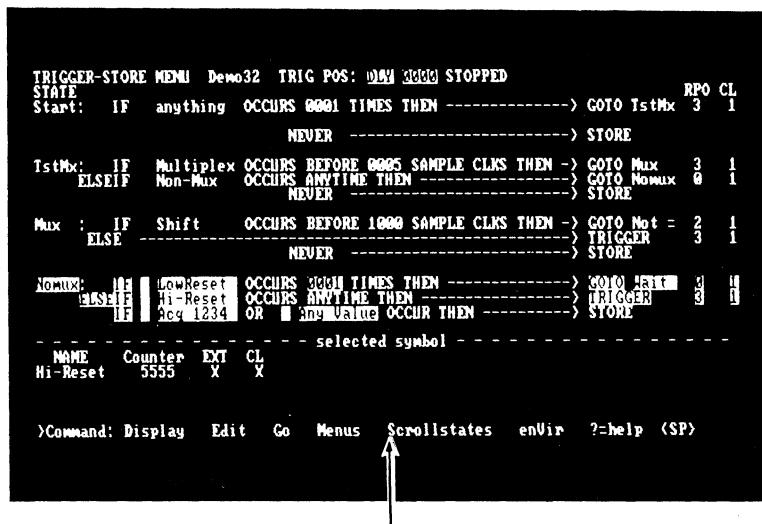
▀▀▀▀ SPECIAL COMMAND-LINE FUNCTIONS

As shown in Figure 5-10, the Trigger-Store Menu provides an additional command-line function that is not available in other menus.

This command, labeled Scrollstates, lets you scroll backward or forward through the defined trigger-store states. To use this command, press S. The display will scroll in the direction indicated by the pointer (< or >) at the front of the command line. To change the direction of the pointer to <, press the < (less than) or − (minus) key. To change the direction of the pointer to >, press the > (greater than) or + (plus) key.

You can also scroll through the trigger-store states by pressing the up and down cursor keys.

00842-036

▪ **Figure 5-10. The Trigger-Store Menu's Scrollstates command.**

████ OVERVIEW

The Trigger-Store Environment Submenu allows you to define several supporting trigger parameters. While these parameters are readily accessible, they have been placed in a submenu so that their meanings are not obscured, and so that the main Trigger-Store Menu is not cluttered.

In this submenu, you can set up Multiple-Preview Acquisition to enhance the trigger's storage qualifier functions. You can specify whether the analyzer automatically stores all state transitions (i.e., the events which cause the analyzer to move from one state to another). Plus, you can establish starting values for the trigger states and the Real-Time Programmable Outputs (RPOs).

In the rest of this subsection, you'll find detailed information about how to use this submenu and its features.

████ SUBMENU FIELD SELECTIONS

The Trigger-Store Environment Submenu is called to the screen by pressing V for enVironment, then T for Trigger-Store. It can be called from any state analyzer menu. When it is called, the submenu temporarily covers the lower half of the screen.
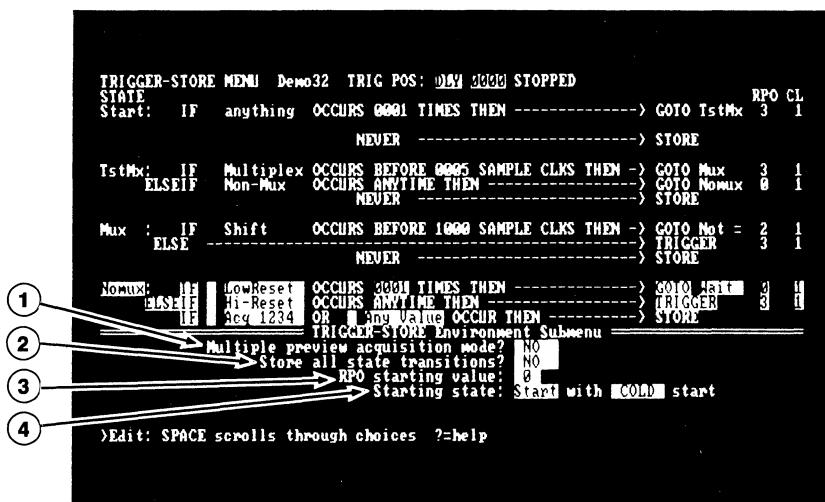
Figure 5-11 illustrates how the submenu appears when it is called from the Trigger-Store Menu. The submenu is automatically in edit mode; it has no command line. You exit the submenu by pressing the escape key.

Use the numbered callouts in Figure 5-11 as a reference while reading the following field descriptions.

00842-037

```
TRIGGER-STORE MENU  Demo32  TRIG POS: ▒▒▒ ▒▒▒▒ STOPPED
STATE                                                              RPO CL
Start:   IF   anything  OCCURS 0001 TIMES THEN -------------> GOTO TstMx  3   1
                        NEVER  -----------------------> STORE

TstMx:   IF   Multiplex OCCURS BEFORE 0005 SAMPLE CLKS THEN -> GOTO Mux    3   1
        ELSEIF Non-Mux   OCCURS ANYTIME THEN ----------------> GOTO Nomux  0   1
                        NEVER  -----------------------> STORE

Mux  :   IF   Shift     OCCURS BEFORE 1000 SAMPLE CLKS THEN -> GOTO Not =  2   1
        ELSE  ---------------------------------------------> TRIGGER       3   1
                        NEVER  -----------------------> STORE

Nomux:   IF   LowReset  OCCURS 0001 TIMES THEN -------------> GOTO Wait    0   0
        ELSEIF Hi-Reset  OCCURS ANYTIME THEN ----------------> TRIGGER      0   0
              IF Acq 1234 OR  Any Value OCCUR THEN ---------> STORE
    =============== TRIGGER-STORE Environment Submenu ===============
              Multiple preview acquisition mode?  NO
              Store all state transitions?  NO
              RPO starting value:  0
              Starting state: Start with COLD start

>Edit: SPACE scrolls through choices  ?=help
```

■ Figure 5-11. Trigger-Store Environment Submenu.

(1) Multiple-Preview Acquisition

This field is used to establish Multiple-Preview Acquisition. Using the space bar, you can select:

■ NO—Multiple-Preview Acquisition is turned off.

■ YES—Multiple-Preview Acquisition is turned on.

When Multiple-Preview Acquisition is turned off, the state analyzer acquires data as specified in the Trigger-Store Menu. It stores events only if they match the specified storage qualifiers and trigger conditions.

When Multiple-Preview Acquisition is turned on, the state analyzer still acquires data as specified in the Trigger-Store Menu, but it also acquires up to seven unqualified events immediately preceding each qualified event. This way, you see a small window of data leading up to the events you are interested in.

☞ NOTE

A detailed application example of how to use Multiple-Preview Acquisition is provided in Section 7 of this manual.

② **Store All State Transitions**

This field lets you specify whether or not the state analyzer acquires all events that cause state transitions (as defined by the Trigger-Store Menu).

Using the space bar, you can select:

- ■ NO—the state analyzer does not automatically acquire events that cause trigger state transitions. It only acquires the events if they match the specified storage qualifier conditions.

- ■ YES—the state analyzer does automatically acquire all events that cause trigger state transitions, regardless of whether or not the events match the specified storage qualifier conditions.

The usefulness of always storing state transitions becomes apparent when you set up NEVER STORE storage qualifiers in the Trigger-Store Menu.

For example, suppose you want to monitor several subroutine accesses, but do not want to store all of the subroutine activity. You could set up a state sequence that follows the paths of the subroutines and select NEVER STORE storage qualification. Storing all state transitions would then allow you to verify that the subroutine accesses occurred correctly, without scrolling through a lot of memory data.

③ **RPO Starting Value**

This field is used to specify the starting values of the Real-Time Programmable Output (RPO) signals.

Because the RPOs change at state transitions, you must define their initial value—the value the RPOs are set to when the state analyzer is started. Then, when the state analyzer leaves the first state, the RPO values change to the value defined in the first state.

In default, the RPOs start with a value of 0. Using the space bar, you can change this starting value to any value listed in Table 5-3.

■ **Table 5-3: RPO Starting Values**

Field Selection	RPO2	RPO1
0	0	0
1	0	1
2	1	0
3	1	1

④ **Starting State**

When you name the starting state in this field, the state analyzer starts operating from any of the 15 defined trigger states. For example, you could define several short trigger sequences to perform individual tests and have all the sequences resident in the state machine simultaneously. You would indicate which test to perform by selecting the state name of the beginning desired state as the start state.

At the end of an acquisition cycle, one data acquisition remains in the analyzer's pipeline. With a WARM start this data will be treated as the first value of the next acquisition, and with a COLD start it will be ignored.

All selections are made by using the space bar.

◼◼◼ OVERVIEW

The Display Menu lets you view acquired data according to the format set the Format Menu. It also enables you to compare reference memory with acquisition memory and view any differences in highlighted video.

The menu provides three basic state displays, including:

- ◼ Acquisition Memory Display
- ◼ Reference Memory Display
- ◼ Acquisition and Reference Memory Comparisons

The rest of this subsection discusses these displays and their associated programmable fields. It also discusses the special command-line functions that allow you to manipulate the memories and their displays.

☑ **NOTE**
Certain menu parameters, such as comparison limits, are set up via the Display Environment Submenu. This submenu will be discussed in the next subsection.

◼◼◼ ACQUISITION MEMORY DISPLAY

Figure 5-12 illustrates a typical state display of acquisition memory. You enter this display by pressing D for Display from the command line.
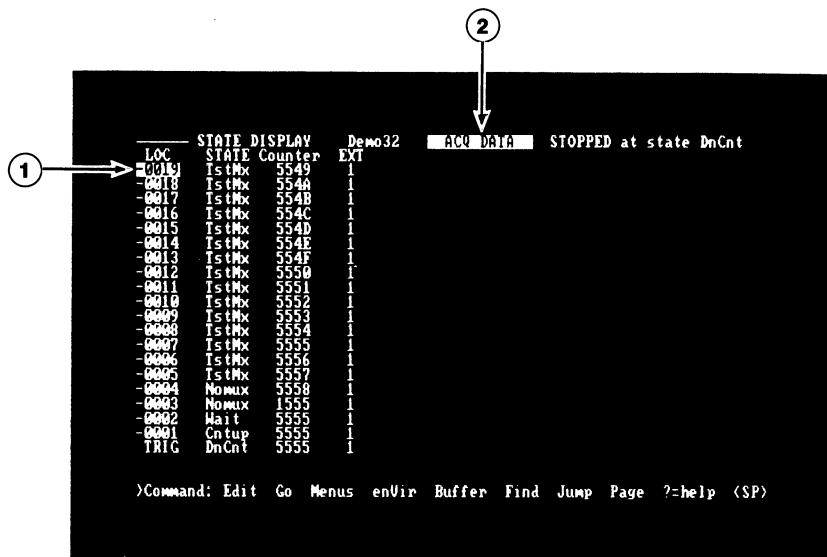
The state display is formatted according to the specifications you have entered in the Format Menu. The column headings reflect the channel group names and their display order. The data is presented in the chosen radix and polarity.

There are only two programmable fields in the Display Menu: one programs the starting memory location for the top of the data list, and the other lets you specify whether to display data from acquisition memory, reference memory, or both.

Use the numbered callouts in Figure 5-12 while reading the following field descriptions. To change a field value, you must be in the edit mode (press E for Edit), and you must have the screen cursor positioned in the field.

If the HELP DISK is installed, you can use the ?=Help function to obtain information about a particular field. To do this, enter the edit mode, then move the screen cursor to the desired field and press the ? (question mark) key.

00842-038

②

```
──────  STATE DISPLAY    Demo32   ▮ ACQ DATA ▮  STOPPED at state DnCnt
   LOC    STATE Counter  EXT
① ──▶ -0019  TstMx   5549   1
      -0018  TstMx   554A   1
      -0017  TstMx   554B   1
      -0016  TstMx   554C   1
      -0015  TstMx   554D   1
      -0014  TstMx   554E   1
      -0013  TstMx   554F   1
      -0012  TstMx   5550   1
      -0011  TstMx   5551   1
      -0010  TstMx   5552   1
      -0009  TstMx   5553   1
      -0008  TstMx   5554   1
      -0007  TstMx   5555   1
      -0006  TstMx   5556   1
      -0005  TstMx   5557   1
      -0004  Nomux   5558   1
      -0003  Nomux   1555   1
      -0002  Wait    5555   1
      -0001  Cntup   5555   1
       TRIG  DnCnt   5555   1

 >Command: Edit  Go  Menus  enUir  Buffer  Find  Jump  Page  ?=help  (SP)
```

■ **Figure 5-12. Acquisition Memory Display.**

① **LOC (Location)**

This field lets you specify the starting memory location for the top of the data list. The trigger position has a number value of 0. Words acquired before the trigger (pre-trigger data) have negative number values, and words acquired after the trigger (post-trigger data) have positive number values.

You can update the display to any portion of memory, by entering the following values into the LOC field:

■ -9999 to +9999—the display updates to the location number specified, or to the location nearest that number. For example, if you specify -9999 and the lowest pre-trigger data location is -23, the display will update to location -23.

■ B—the display updates to the Beginning location in memory.

■ E—the display updates to the Ending location in memory.

■ T—the display updates to the Trigger's location in memory.

Once you have entered the location value, press the return key to complete the entry.

> ☑ **NOTE**
> This is only one method for moving through the data display. Other methods are described later in this subsection, under *Special Command-Line Functions.*

(2) **Memory Selection**

This field lets you select the memory display. Using the space bar, you can select:

- ACQ—the display shows the current acquisition memory.

- REF—the display shows the current reference memory.

- <—ACQ REF—>—the display shows an acquisition and reference memory comparison.

Figure 5-12 illustrates an ACQ memory display. A REF memory display and an <—ACQ REF—> memory comparison are illustrated in Figures 5-13 and 5-14, respectively.

▦ REFERENCE MEMORY DISPLAY

Figure 5-13 illustrates a typical display of reference memory. To enter this display, select the REF value in the Memory Selection field.

> ☑ **NOTE**
> To achieve a display similar to that shown in Figure 5-13, you must first enter data into reference memory. To do this, use the Ref<—acq command (see *Special Command-Line Functions* later in this subsection).

The numbered callouts in Figure 5-13 illustrate the programmable fields within the reference memory display. Only the field numbered 3 is discussed here, since it is the only field different than those provided in the acquisition memory display. For information on how to manipulate fields numbered 1 and 2, refer back to the *Acquisition Memory Display* portion of this subsection.

00842-039

■ **Figure 5-13. Reference Memory Display.**

③ Editing Reference Data

These fields let you modify the content of reference memory. You can change any data value as long as you use the appropriate radix notation. The only exceptions are:

1. You cannot edit any values in the STATE group.

2. You cannot edit values with an ASCII radix.

To scroll the reference data window, use the LOC field or use any of the command-line functions listed later in this subsection.

■ MEMORY COMPARISONS

Figure 5-14 illustrates a typical comparison display of the acquisition and reference memories. The two memories are presented side by side, with acquisition memory on the left side of the display and reference memory on the right.

00842-040

■ Figure 5-14. Acquisition and Reference Memory Comparison.

Comparison of Multiple-Preview Acquisitions. If either set of data was acquired in the Multiple-Preview Acquisition mode, there may be some blank lines in the data (see Figure 5-15). A blank line indicates a gap in the sampled data; two samples separated by a blank line were not acquired on successive sample clocks.

When you're comparing acquisition and reference data, the blank lines may not occur in the same places in both sets of data. In this case a blank line will appear in the data that contains a gap, and a set of vertical bars will appear in the data that is continuous.

00842-041

■ Figure 5-15. Comparison of Multiple-Preview Acquisition data.

■■■ SPECIAL COMMAND-LINE FUNCTIONS

In addition to the standard commands, the Display Menu provides several specialized commands that allow you manipulate the display of data.

These commands do not all fit on the command line. To view all of the commands, press the space bar.

The following paragraphs describe the special commands in the order they appear on the command line. The bold, capitalized letter in a command indicates that you must press that letter to execute the command. Brackets < > around a command indicate that you must press the corresponding key to execute the command.

< or >   The pointer at the front of the command line controls the direction of paging and data find functions.

A < pointer specifies that paging and find functions will move backward through memory (toward pretrigger data). To set the pointer in this direction, press the < (less than) or − (minus) key.

5-45

A > pointer specifies that paging and find functions will move forward through memory (toward post-trigger data). To set the pointer in this direction, press the > (greater than) or + (plus) key.

Buffer This command lets you store a data line in a special buffer for later unbuffering in the Symbol Menu. The data line stored is indicated at the top of the display by the reverse-video LOC field.

Find This command allows you to search acquisition or reference memory for a specific data pattern. The pointer at the front of the command line controls the direction of the search.

When the acquisition and reference memories are both displayed, the Find command searches acquisition memory for the pattern.

☑ **NOTE**
The data pattern being searched is specified in the Display Environment Submenu.

Next_diff This command searches acquisition and reference memory for the next difference between the two. The pointer at the front of the command line controls the direction of the search.

Total_diff This command prints a line on the bottom of the display that lists the total number of differences between acquisition and reference memory. The line also lists the location of the first difference in memory, and the location of the last difference.

Page This command allows you to move through the state data one page at a time. The pointer at the front of the command line determines the direction of the paging.

Ref<—acq This command transfers acquisition memory to reference memory. The previous contents of reference memory are lost.

☑ **NOTE**
The boundaries of this memory transfer are set up in the Display Environment Submenu.

Jump	This command lets you jump to specific locations within memory. These locations include:
	Begin —jump to the beginning of memory. **E**nd —jump to the end of the memory. **T**rigger —jump to the trigger's position in memory.
<Home>	This key moves the display to the beginning of memory.
<End>	This key moves the display to the end of memory.
<PgUp>	This key moves the display one page backward in memory (toward pre-trigger data).
<PgDn>	This key moves the display one page forward in memory (toward post-trigger data).

▄▄▄ OVERVIEW

Through the Display Environment Submenu, you may specify control parameters that will determine how data is acquired and displayed. These include:

- Setting acquisition (Go) modes.

- Setting the display mode to standard or compressed.

- Setting up comparisons between acquisition and reference memory, with options for range, offset, and mask.

- Setting up the find function, and requesting either that all states be included in the find pattern search or that the find pattern occur in a specified state.

The rest of this subsection shows you how to set up and use the above parameters, and explains the affects of these parameters on the state display.

▄▄▄ SUBMENU FIELD SELECTIONS
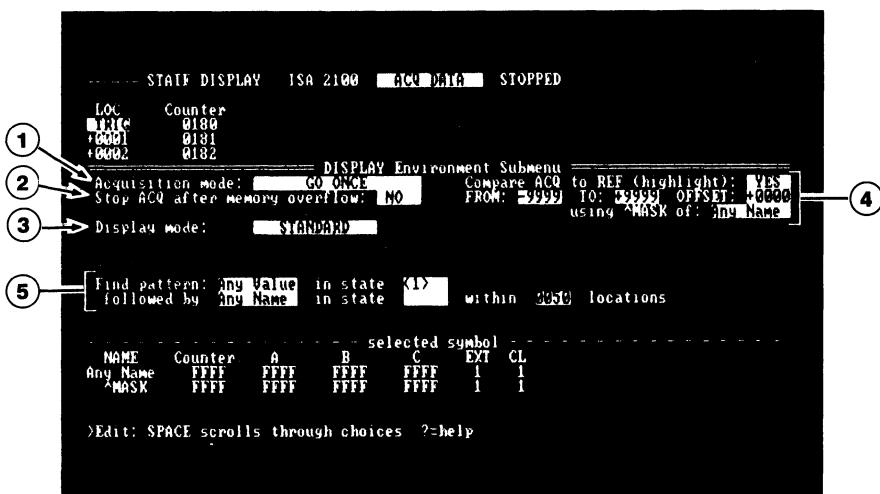
The Display Environment Submenu is called to the screen by pressing V for enVironment, then D for Display from the command line. It can be called from any state analyzer menu. When it is called, the submenu temporarily covers the bottom half of the screen.

Figure 5-16 illustrates how the submenu appears when it is called from the Display Menu. The submenu is automatically in edit mode; it does not have a command line. You exit the submenu by pressing the escape key.

Use the numbered callouts in Figure 5-16 as a reference while reading the following field descriptions.

00842-042

```
------ STATE DISPLAY   ISA 2100  ■ ACQ DATA ■  STOPPED

    LOC      Counter
  ■ TRIG      0180
  +0001       0181
  +0002       0182
===================== DISPLAY Environment Submenu ====================
Acquisition mode:        GO ONCE         Compare ACQ to REF (highlight):  YES
Stop ACQ after memory overflow:  NO      FROM: -9999  TO: -9999  OFFSET: +0000
                                              using ^MASK of: Any Name

Display mode:           STANDARD


 Find pattern: Any Value  in state  <1>
 followed by  Any Name   in state        within  0050  locations


- - - - - - - - - - - - - - - selected symbol - - - - - - - - - - - - - - - - -
  NAME     Counter   A       B       C     EXT  CL
Any Name    FFFF    FFFF    FFFF    FFFF    1    1
  ^MASK     FFFF    FFFF    FFFF    FFFF    1    1

>Edit: SPACE scrolls through choices  ?=help
```

① ② ③ ⑤ ④

■ **Figure 5-16. The Display Environment Submenu.**

① Acquisition Mode

This field establishes the state analyzer's acquisition (Go) mode. Using the space bar, you can select:

■ GO ONCE— the analyzer acquires data until the trigger condition and delay count, if any, have been met.

■ GO FOREVER— the analyzer makes repetitive acquisitions, using the starting state specified in the Trigger-Store Environment Submenu.

■ GO TIL ACQ = REF—the analyzer makes repetitive acquisitions until the acquisition and reference memories are equal.

■ GO TIL ACQ < > REF—the analyzer makes repetitive acquisitions until the acquisition and reference memories are unequal.

When the state analyzer is running in the above modes, you can stop acquisition in one of two ways. Press H (Halt) to stop the analyzer immediately; or press S (Stop) to stop the analyzer after the next trigger condition has been satisfied.

5-49

② Stop ACQ After Memory Overflow

This field is useful when you cannot specify an accurate or appropriate trigger condition. It allows you to stop acquisition when the analyzer detects a memory overflow condition.

Using the space bar, you can select:

■ NO—the analyzer does not check for a memory overflow condition. Acquisition stops when a trigger has been generated and the post-trigger delay has been satisfied.

■ YES—the analyzer periodically checks for a memory overflow condition. If an overflow has occurred, the analyzer will then stop acquisition, regardless of whether or not a trigger has been generated.

☑ **NOTE**
A YES value in this field does not prevent data from being overwritten in the state analyzer's memory; it just stops acquisition the first time the analyzer detects that overwriting has occurred.

③ Display Mode

This field is used to compress the state display. Using the space bar, you can select:

■ STANDARD—data is displayed in the standard format.

■ COMPRESSED—data is displayed in the compressed format.

The compressed format packs more data onto the display by shortening group names, removing spaces, and creating additional columns of data display. In a system with a large number of channels or several groups with binary radix, it may not be possible to display more than one column of data in compressed format.

Figure 5-17 illustrates an example of the compressed display format.

00842-043

```
          STATE DISPLAY    Demo32    ACQ DATA    STOPPED at state DnCnt
     LOC   STATE Count EX     LOC   STATE Count EX     LOC   STATE Count EX
    -0059  TstMx 5521  1    -0039  TstMx 5535  1    -0019  TstMx 5549  1
    -0058  TstMx 5522  1    -0038  TstMx 5536  1    -0018  TstMx 554A  1
    -0057  TstMx 5523  1    -0037  TstMx 5537  1    -0017  TstMx 554B  1
    -0056  TstMx 5524  1    -0036  TstMx 5538  1    -0016  TstMx 554C  1
    -0055  TstMx 5525  1    -0035  TstMx 5539  1    -0015  TstMx 554D  1
    -0054  TstMx 5526  1    -0034  TstMx 553A  1    -0014  TstMx 554E  1
    -0053  TstMx 5527  1    -0033  TstMx 553B  1    -0013  TstMx 554F  1
    -0052  TstMx 5528  1    -0032  TstMx 553C  1    -0012  TstMx 5550  1
    -0051  TstMx 5529  1    -0031  TstMx 553D  1    -0011  TstMx 5551  1
    -0050  TstMx 552A  1    -0030  TstMx 553E  1    -0010  TstMx 5552  1
    -0049  TstMx 552B  1    -0029  TstMx 553F  1    -0009  TstMx 5553  1
    -0048  TstMx 552C  1    -0028  TstMx 5540  1    -0008  TstMx 5554  1
    -0047  TstMx 552D  1    -0027  TstMx 5541  1    -0007  TstMx 5555  1
    -0046  TstMx 552E  1    -0026  TstMx 5542  1    -0006  TstMx 5556  1
    -0045  TstMx 552F  1    -0025  TstMx 5543  1    -0005  TstMx 5557  1
    -0044  TstMx 5530  1    -0024  TstMx 5544  1    -0004  Nomux 5558  1
    -0043  TstMx 5531  1    -0023  TstMx 5545  1    -0003  Nomux 1555  1
    -0042  TstMx 5532  1    -0022  TstMx 5546  1    -0002  Wait  5555  1
    -0041  TstMx 5533  1    -0021  TstMx 5547  1    -0001  Cntup 5555  1
    -0040  TstMx 5534  1    -0020  TstMx 5548  1     TRIG  DnCnt 5555  1

    )Command: Edit  Go  Menus  enVir  Buffer  Find  Jump  Page  ?=help  (SP)
```

■ **Figure 5-17. The compressed display format.**

④ **Compare ACQ to REF**

These fields control the acquisition and reference memory comparisons. They specify display highlighting, comparison limits, comparison offset, and masking.

Highlighting. You can turn display highlighting on and off using the space bar. Your choices include:

■ NO—the highlighting is turned off.

■ YES—the highlighting is turned on.

When you select YES, any differences between acquisition and reference data that are not masked by the compare mask symbol will be highlighted in the Display Menu.

From/To. The from and to parameters determine the range of data locations for all comparisons between acquisition and reference memories. These boundaries are also used for the transfer of acquisition data to reference memory.

The boundaries for the comparison can range from -9999 to +9999.

☑ **NOTE**
If you are using the μAnalyst's 256K SYSTEM DISK, the size of reference memory may be limited. For details, refer to the front of this manual and the page titled *Software Versions Covered By This Manual.*

Offset. The offset parameter allows you to skew the reference data in relation to acquisition data. An offset of one or more will move the data in reference memory down (or forward) on the Display Menu. The offset parameter will be ignored during a transfer of acquisition to reference memory.

^ Mask. The ^ Mask parameter is the name of a symbol defined in the Symbol Menu. The mask part of the symbol specified will be used as a mask during compare operations. These operations include highlighting, GO TIL ACQ = REF or GO TIL ACQ < > REF acquistion modes, and Next_Diff and Total_Diff commands.

A zero bit in the ^ Mask field means don't compare for that channel.

Symbols are entered into the ^ Mask field by typing the selected symbol's name. If you cannot remember all of the symbol names within the Symbol Menu, you can use the Control-S (^ S) key to scroll through the symbol names. Each symbol name and its value appears on the bottom of the screen under the heading "selected symbol."

Operational Notes. For all the compare functions, the STATE group is ignored. When acquisition and reference data do not overlap, they are considered equal. For example, if the amount of reference data differs from the amount of acquired data, all data in unmatched line numbers will be considered equal (subject, however, to the offset parameter).

⑤ **Find Pattern**

This field establishes the find pattern used by the Display Menu's Find command.

The find pattern may consist of from one to four words, with each word specified by its name in the Symbol Menu. Note that the mask part of each symbol may be used to ignore part of the symbol during a find operation. A zero bit in the symbol's mask is a don't compare for that channel.

Use the insert key to insert a word in the search pattern. Use the delete key to delete a word in the search pattern.

Each word in the search pattern has a corresponding state field. This field lets you specify which trigger-store state the word must be in to match the find pattern. For example, suppose you set up the following conditions:

Find pattern Multiplex in state Mux

In this case, the Display Menu will compare the Multiplex pattern to all state samples that were acquired in the state Mux. Any other state samples will be ignored.

You can select between the various trigger-store states by using the space bar. If you leave the state field set to all blanks, the find word will be compared to all trigger-store states.

If the find pattern consists of more than one word, each word in the pattern may be separated by from 1 to 4096 locations. A value of 1 means that the words must be found in consecutive locations (i.e., there can be no other samples between them).

For example, suppose you set up the following conditions:

Find pattern	Multiplex	in state	Mux	
followed by	Non-Mux	in state	Nomux	within 01 locations
followed by	LowReset	in state	(blank)	within 50 locations

When you execute the Find command, the Display Menu will first look for the word Multiplex in state Mux. If this word is found, the menu will then look at the next immediate memory location to see if it contains the word Non-Mux in state Nomux.

If the next immediate location does not contain Non-Mux, the menu will again look for Multiplex followed by Non-Mux, and so on. If the next immediate location does contain Non-Mux, the menu will look for LowReset in the next 50 locations.

If LowReset is found, the menu will update the display to that location. If LowReset is not found, the menu will again search for Multiplex, followed by Non-Mux, and so on.

The Display Menu will continue to search for the pattern throughout the entire state memory. If the pattern cannot be found, a message will appear on the display.

■■■ OVERVIEW

The Histogram Menu is used to perform range-activity analysis on acquired state data. This can include address activity or specific data activity, optionally qualified by some value you select.

With this menu, you can define up to 16 ranges for a particular channel group. The system then calculates and displays the percentage of activity occurring within each range. You can view the range activity levels in a proportional bar graph, or you can view an occurrence count of the events falling into each range.

A defined range is called a bin, and the process of determining what data or address falls into a bin is called binning.

The rest of this subsection illustrates how you use the features of the Histogram Menu.

■■■ MENU FIELD SELECTIONS

Figure 5-18 illustrates a typical Histogram Menu display. To enter this menu, press M for Menu, then H for Histogram from the command line.
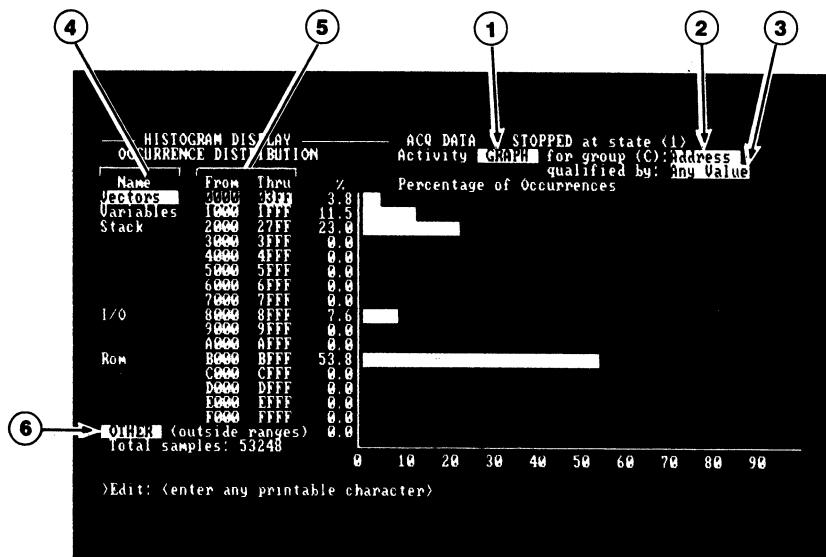
Use the numbered callouts in Figure 5-18 as a reference while reading the following field descriptions. To change a menu field, you must be in the edit mode (press E for Edit), and you must have the screen cursor positioned in the field.

If the μAnalyst HELP DISK is installed, you can use the ?=Help function to obtain information about a particular field. To do this, enter the edit mode, then move the screen cursor to the desired field and press the ? (question mark) key.

00842-044

■ **Figure 5-18. The Histogram Menu and its fields.**

(1) Activity

This field determines the menu's display format. Using the space bar, you can select:
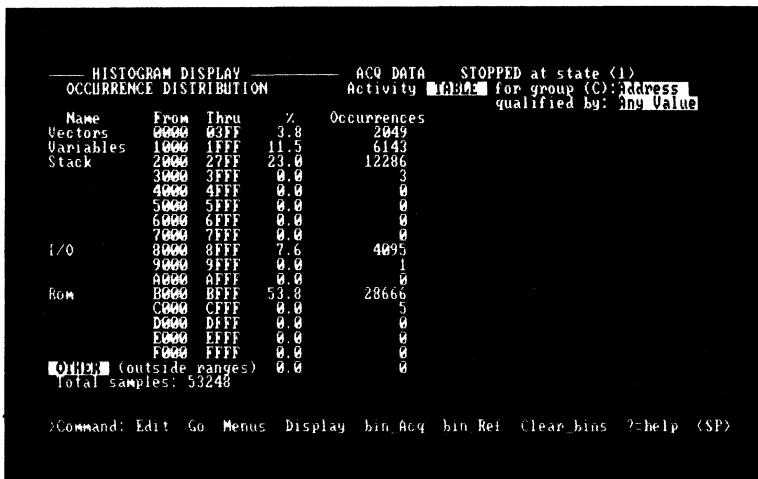
- ■ GRAPH—the activity levels within the ranges are displayed as a proportional bar graph.

- ■ TABLE—the activity levels within the ranges are displayed as a total occurrence count.

Figure 5-18 illustrates how the activity levels appear when a graph display is selected. Figure 5-19 illustrates how these same activity levels appear when a table display is selected.

00842-045

```
──── HISTOGRAM DISPLAY ────        ACQ DATA    STOPPED at state (1)
   OCCURRENCE DISTRIBUTION          Activity ▉TABLE▉ for group (C): Address
                                             qualified by: Any Value
   Name      From  Thru    %    Occurrences
Vectors      0000  03FF   3.8       2049
Variables    1000  1FFF  11.5       6143
Stack        2000  27FF  23.0      12286
             3000  3FFF   0.0          3
             4000  4FFF   0.0          0
             5000  5FFF   0.0          0
             6000  6FFF   0.0          0
             7000  7FFF   0.0          0
I/O          8000  8FFF   7.6       4095
             9000  9FFF   0.0          1
             A000  AFFF   0.0          0
Rom          B000  BFFF  53.8      28666
             C000  CFFF   0.0          5
             D000  DFFF   0.0          0
             E000  EFFF   0.0          0
             F000  FFFF   0.0          0
▉OTHER▉ (outside ranges)  0.0          0
   Total samples: 53248

>Command: Edit  Go  Menus  Display  bin_Acq  bin_Ref  Clear_bins  ?=help  (SP)
```

■ **Figure 5-19. Histogram Menu with a table display.**

② **For Group**

This field selects the channel group that is to be analyzed. Only the data acquired from one channel group can be binned at any one time. The organization of channel groups is determined in the Format Menu.

Note that as you scroll through the groups, the names originally assigned in the Format Menu appear in the reverse video field. You cannot change groups names here, but you may go back to the Format Menu and do so. If no group name was assigned in the Format Menu, the default group name (A, B, C, etc.) appears in the field. The single letter reference always appears parenthetically just before the reverse video field.

Also note that the numerical range of the group, as determined by its assigned channels, is reflected in the From and Thru fields (see menu callout number 5).

③ Qualified By

This field lets you select a qualifier from the current list of symbols in the Symbol Menu. All values will be binned if the selected qualifier symbol is set to all don't cares (i.e., the Any Value symbol). Alternately, you can selectively use don't cares to range the data or look for a specific type of data.

For example, you might bin address information qualified by memory reads. The qualifier symbol would select the control line pattern that indicates a memory read, and the group containing the address bus information would be binned.

For more information on setting symbol values, refer to the Symbol Menu.

④ Name

These fields are used for naming the ranges. You can enter any name up to 10 characters. The names are for your reference only and do not affect the binning process. The up and down cursor keys select which line to edit.

⑤ From—Thru

These fields establish the inclusive beginning and ending limits for the ranges. Ranges may overlap, but the data will fall into the first bin that satisfies the range specification. Data will not be duplicated in any subsequent bins, even though another bin may satisfy the range requirements. The up and down cursor keys select which line to edit.

☞ **NOTE**

To specify a range of one value, enter the same value into the From and Thru fields.

⑥ Other (Outside Ranges)

This field is used to include or exclude data that does not fall within any of the specified ranges. Using the space bar, you can select:

■ OTHER—bin the data falling outside the ranges in this group.

■ A blank field—ignore the data falling outside the ranges.

When OTHER is selected and its contents are greater than zero, percentages of the other ranges may change.

■■■■ OPERATIONAL NOTES

Trigger-Store Setups for Range-Activity Measurements.
When you are acquiring data for a range-activity measurement, try to
capture as broad a sampling as possible. The more code sampled
during a given acquisition, the better the statistical distribution.

One way to optimize the state analyzer's sampling capability
is to set up a storage qualification loop that captures a selection of
the executing code rather than every cycle.

For example, you could set up the following storage loop in
the Trigger-Store Menu:

<1> IF <Any Value> OCCURS AFTER 0050 SAMPLE CLOCKS THEN
       —> GOTO <2>
     NEVER———————————————————————————> STORE

<2> IF <Any Value> OCCURS 0001 TIMES THEN———> GOTO <1>
     ALWAYS——————————————————————————> STORE

This storage loop stores one out of every 51 cycles. The state
analyzer is able to monitor a much broader range of code before bin-
ning the data into the histogram ranges.

Notice that the above storage loop does not establish a trig-
ger condition. To stop acquisition in this case, use the Stop ACQ After
Memory Overflow option in the Display Environment Submenu.

Printing the Histogram Display. The bins must be cleared
before binning a new group. You will be prompted if you attempt to
change bin groups without first clearing the bins. If you do not wish
to clear the bins, press N for No when the prompt appears. If you
wish to clear the bins, enter Y for Yes.

Before clearing the bins, you may wish to first print the dis-
play using the PrtSc (PrintScreen) key. Bin names and range values
can be saved on disk as part of a setup file in the I/O menu, but the
binned data cannot be saved.

■■■■ SPECIAL COMMAND-LINE FUNCTIONS

In addition to the standard command functions, the Histogram Menu provides specialized commands that allow you to control the binning process.

These commands do not all fit on the command line. To view the entire list of commands use the space bar.

The following paragraphs describe the commands in the order they appear on the command line. The bold, capitalized letter in each command indicates the letter key you must press to activate the command.

Go

This command starts the acquisition and automatically bins acquisition memory for the range and group specified. This can be most useful when one of the continuous Go modes is used (as specified in the Display Environment Submenu). Repeated acquisitions are made, thus increasing the number of observations.

Go with automatic binning only occurs when G is pressed while the Histogram Menu is on the screen.

bin_**A**cq

This command bins the data from acquisition memory. If data was previously binned and the bins were not cleared, the results of the new binning are added to the old bin data.

bin_**R**ef

This command bins the data from reference memory. If data was previously binned and the bins were not cleared, the results of the new binning are added to the old bin data.

Clear_bins

This command clears the bins.

OVERVIEW

The I/O Menu lets you save menu setups and data. You can load the setups and data onto disk for mass storage, or you can output them to an IBM-compatible printer.

Setups and data stored on disk can be recalled and loaded into the state analyzer at any time. Setup files contain information from the Format, Symbol, Trigger-Store, and Histogram Menus, and from the Trigger-Store and Display Environment Submenus. Data files contain the data contents from reference memory.
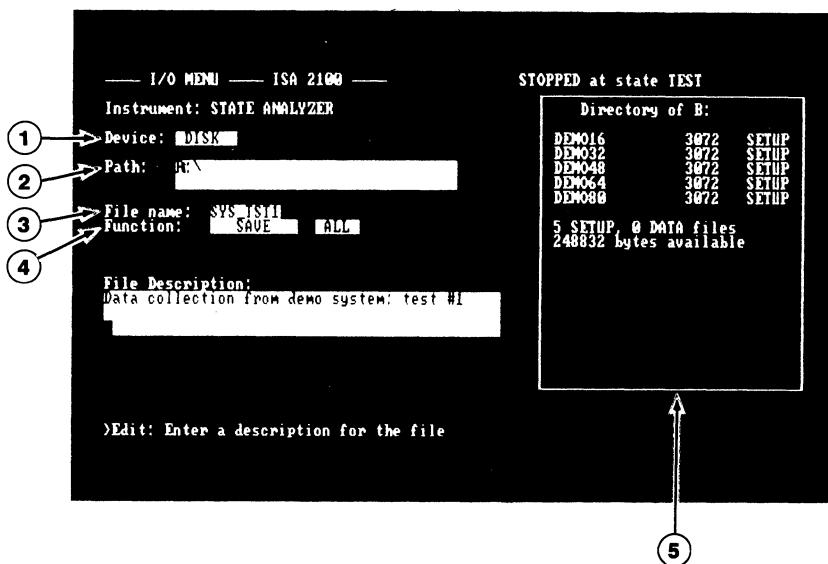
Using the menu's printer function, you can also make hard copies of the Symbol, Trigger-Store, and Display Menus. You can send these menus to a printer or save them on disk for later output to a printer. The rest of this subsection shows how to use the I/O Menu and its disk-storage and printer capabilities.

MENU FIELD SELECTIONS FOR DISK

Figure 5-20 illustrates the I/O Menu when DISK is the selected device. You enter the menu by pressing M for Menu, then I for I/O from the command line. You select the DISK device by using the space bar in the Device field (see menu callout number 1).

Use the numbered callouts in Figure 5-20 as a reference while reading the following field descriptions. To program a menu field, you must be in the edit mode (press E for Edit), and you must have the screen cursor positioned in the field.

If the HELP DISK is installed, you can use the ?=Help function to obtain information about any particular field. To do this, enter the edit mode, then move the screen cursor to the desired field and press the ? (question mark) key.

00842-046

■ Figure 5-20. The I/O Menu with DISK as the selected device.

① Device

This field establishes which I/O device the state analyzer will use: DISK or PRINTER. You make the selection by using the space bar.

☞ NOTE

For information on how to use the PRINTER device, see Figure 5-21 later in this subsection.

② Path

This field lets you specify the drive-directory path that leads to the file you want to use during the I/O function.

Initially, the Path field is set to the default drive and its root directory. You can set the field to an alternate path using up to 64 characters.

☞ NOTE

Only one path name can be specified at any one time. The state analyzer will not search more than one path.

The delimiters of the path name are as follows:

<d>: \ directory1 \ directory2 \ . . .

where <d>: is a drive location. If you do not want to change drives, leave out the <d>: designation and just enter the directories. The state analyzer will search for the directories in the current drive.

When a new drive or directory is entered into the Path field, the menu's directory window is automatically updated to show the state analyzer files located at that directory level (see callout number 5). The window will not show any DOS system files or user programs.

③ File Name

This field specifies which file will be used during the I/O function. For SAVE functions, enter a new file name. For LOAD or DELETE functions, enter the name of a file that already exists in the current directory.

A file name can be up to eight characters in length. Certain characters cannot be used in the name, such as dashes, commas, dollar signs, and others. The message line at the bottom of the menu will advise you when an illegal character has been entered.

④ Function

This field specifies the I/O function to be performed on the selected file. Using the space bar, you can specify one of the functions listed below:

■ SAVE—saves the file on disk.

■ LOAD—loads the file into the state analyzer.

■ DELETE—deletes the file from disk.

The following paragraphs describe each of these functions. You execute a given function by pressing X for eXecute from the command line. (Refer to *Special Command-Line Functions* later in this subsection.)

SAVE Function. This function lets you save a new file on disk. The new file is identified by three elements: a file name, a file type, and a description.

☞ **NOTE**
The disk used for storing the new file must be formatted for DOS (version 2.00 or higher). For procedures on how to format a diskette for DOS, refer to your DOS manual.

The file name is entered in the File Name field as described under callout number 3. If the entered name is the name of an already existing file, the existing file will be overwritten with the new file. In this case, the μAnalyst will ask you to confirm the action by pressing Y for Yes.

The file type is specified in the field which appears next to the SAVE field. Using the space bar, you can select:

- SETUP—which saves menu setup information.

- DATA—which saves the data from reference memory. (Data from acquisition memory cannot be transferred directly onto disk. To save acquisition memory, you must first transfer it to reference memory.)

☑ **NOTE**
The byte structure used for DATA files is provided in Appendix B at the back of this manual.

- ALL—which saves both the menu setups and the data from reference memory. This selection creates two files with the same name, one for setup and one for data.

The file description is entered into the field that appears below the Function field. You can enter up to 135 characters. The description is saved with the file, and it will reappear on the screen whenever the file is loaded into the state analyzer.

LOAD Function. This function is the reverse of SAVE. It lets you load a file into the state analyzer. The file is identified by its name and type.

When you load a file, the current analyzer setup and/or reference memory will be overwritten by the file. The μAnalyst will ask you to confirm the action by pressing Y for Yes.

DELETE Function. This function is used to delete a file. The file is identified by its name and type.

When deleting a file, the μAnalyst will first display a description of the file, then it will ask you to confirm the action by pressing Y for Yes.

⑤ Directory Window

The directory window lists the state analyzer files that are located at the current drive-directory level.

The files are identified by their name, their byte size, and their type (either SETUP or DATA). The ALL file types are not listed, since they are made up of SETUP and DATA files.

The bottom two lines of the directory window tell you the total number of files contained in the directory and indicate how many bytes are still available for use.
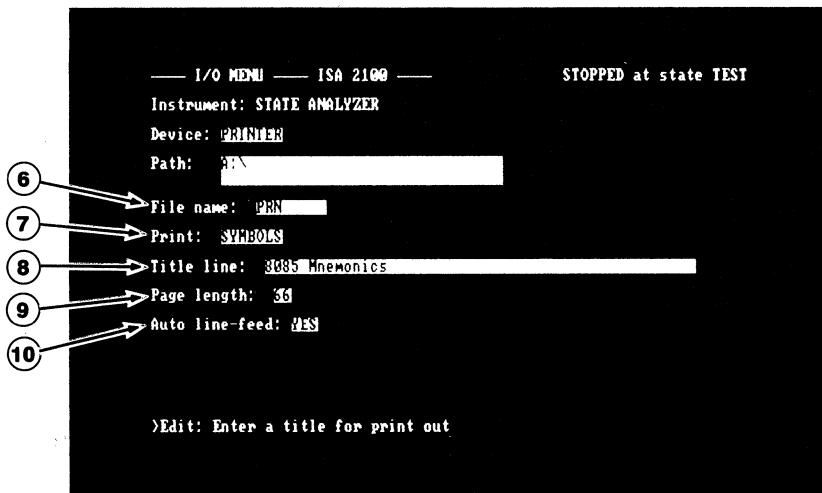
If the directory contains more files than can fit in the window, you can scroll through the files by pressing P for Page_directory from the command line. (Refer to *Special Command-Line Functions* contained later in this subsection).

■■■ MENU FIELD SELECTIONS FOR PRINTER

Figure 5-21 illustrates the I/O Menu when PRINTER is the selected device. You select the PRINTER device by using the space bar in the Device field.

Use the numbered callouts in Figure 5-21 as a reference while reading the following field descriptions. The numbered callouts for this figure do not start at 1, but rather they progress from the last number used under the DISK field selections. This is to avoid confusion between the two displays.

00842-047

■ **Figure 5-21. The I/O Menu with PRINTER as the selected device.**

⑥ File Name

This field specifies the type of print operation. You can send a file to a printer, or you can save an ASCII file on disk for later output to a printer.

To send a file to a printer, enter one of the following reserved DOS device names:

LPT1 or PRN—First Parallel Printer
LPT2—Second Parallel Printer
LPT3—Third Parallel Printer
COM1—Serial Port 1
COM2—Serial Port 2

☑ NOTE

For a complete list of printers that can be used with the μAnalyst, refer to the page titled *PC Compatibiity Requirements* at the front of this manual.

To save a file on disk, enter any name up to eight characters. The new file will be stored at the drive-directory location specified in the Path field.

When you save a file on disk, the new file is only suitable for later outputting to a compatible printer. The ASCII format used in the file is not suitable for loading into the state analyzer. To create a file that can be loaded into the state analyzer, you must use the DISK device as shown in Figure 5-20.

⑦ Print

This field selects the source of the data to be printed. Using the space bar, you can select:

- ■ SYMBOLS—prints the contents of the Symbol Menu.

- ■ STATES—prints the contents of the Trigger-Store Menu.

- ■ DATA—prints data based on the last-used Display Menu settings, e.g., ACQ, REF, < —ACQ REF— >, standard or compressed.

When you select DATA, you will be prompted to specify the range of data you wish to print: from what point in memory to what point in memory. The message line at bottom of the display will tell you the beginning and ending sequences in memory. Instead of entering numerical ranges, you may also enter B for Beginning, E for End, or T for Trigger.

⑧ Title Line

This field allows you to enter a 50-character page header. The header then appears at the top of each printed page. Each printed page is also automatically numbered.

⑨ Page Length

This field defines the length of the printed page, including title block.

Acceptable values are from 20 to 99 vertical lines. Select the value that fits the physical size of the paper you are using, as well as the lines-per-inch (lpi) setting of your printer.

⑩ Auto Line-Feed

An auto line-feed can be generated after every carriage return. Using the space bar, you can select: YES, generate line-feeds after every carriage return; or NO, do not generate line-feeds.

████ SPECIAL COMMAND-LINE FUNCTIONS

The following paragraphs describe the command-line functions specific to the I/O Menu. The bold, capitalized letter in a command indicates that you must press that letter to execute the command.

Page_directory This command only appears on the command line when the menu is set to the DISK device. The command lets you page forward through the files listed in the directory window. When the last file is reached, the paging starts over from the beginning of the directory.

Page_printer This command only appears on the command line when the menu is set to the PRINTER device. The command lets you send a form feed character (0C, hexadecimal) to the printer.

e**X**ecute This command appears the when menu is set to either the DISK or the PRINTER device. The command executes the I/O function designated on the screen.

Using this section. This section shows you how to use the Model 2100 Interactive State Analyzer in time-aligned acquisition with the Model 2200 Interactive Timing Analyzer.

During this section, it is assumed that you are already familiar with the operating menus of both analyzers. If not, refer to Section 5 of this manual for information on state analyzer menus, and refer to the *Interactive Timing Analyzer Users Manual* for information on timing analyzer menus.

■■■■■ SECTION CONTENTS

■■■ OVERVIEW

You can use the Interactive State Analyzer in conjunction with the Interactive Timing Analyzer if you have both products installed in the μAnalyst 2000 Mainframe. Many combinations of aligned and non-aligned acquisition modes are available for operating the two analyzers together.

Aligned acquisition means that common time reference information is stored with the state and timing data as it is placed in acquisition memory. This allows the synchronous data from the state analyzer to be correlated with the asynchronous data from the timing analyzer, and vice versa.

The correlated state and timing data is shown in a split-screen display. You can view of the same point in time from the different perspectives of the state and timing analyzers. Dual scrolling allows you to move between the analyzer memories simultaneously. When you scroll the state or timing window, the other window keeps in step.

The time correlation between state and timing data is handled internally by the μAnalyst. Post-acquisition software matches the state clocks to the timing acquisition data and uses the Crosslink (CL) signal as a common reference point.

Non-aligned acquisition means that the Crosslink (CL) signal is used in a more general way. Because the acquisitions are not time-aligned for display, more programming flexibility is allowed than in one of the aligned modes.

The options for operating the state and timing analyzers together allow for the state analyzer to arm the timing analyzer, and vice versa.

It is important to note the difference between triggering and arming. Triggering occurs when an incoming sample satisfies the programmed trigger condition. Arming an analyzer does not mean that the analyzer has been triggered, it only means that the analyzer has been enabled to look for its trigger. When the analyzer is not armed, the trigger condition may occur, but the analyzer will ignore it.

The Crosslink (CL) is an internal bus line that is used to communicate the arm/not-armed condition. The CL can be used in either direction—state arms timing or timing arms state. (The signal on the CL line also appears as an output at the CL BNC connector located on the mainframe's back panel).
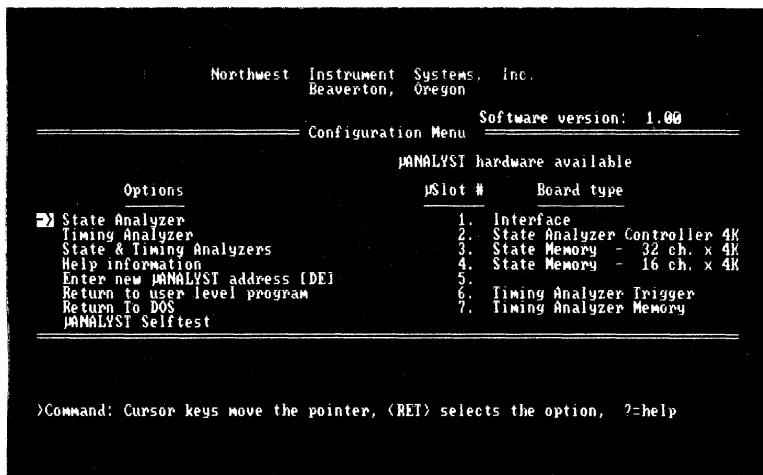
■■■■ ALIGNED/NON-ALIGNED MODES

You select the aligned/non-aligned acquisition modes in the Configuration Menu. As shown in Figure 6-1, a special operating option, titled State & Timing Analyzers, is provided when both analyzers are installed in the μAnalyst mainframe.

To select the State & Timing Analyzers option, move the screen pointer to that option's location, then press the return key. The Configuration Menu will immediately display the aligned/non-aligned acquisition modes, as shown in Figure 6-2.

00842-048

```
                    Northwest  Instrument  Systems,  Inc.
                    Beaverton,  Oregon
                                              Software version:  1.00
                           ===============Configuration Menu ===============

                                     μANALYST hardware available

              Options                      μSlot #      Board type
              -------                      -----        ----------
   ■》 State Analyzer                          1.  Interface
      Timing Analyzer                          2.  State Analyzer Controller 4K
      State & Timing Analyzers                 3.  State Memory  -  32 ch. x 4K
      Help information                         4.  State Memory  -  16 ch. x 4K
      Enter new μANALYST address [DE]          5.
      Return to user level program            6.  Timing Analyzer Trigger
      Return To DOS                           7.  Timing Analyzer Memory
      μANALYST Selftest


   >Command: Cursor keys move the pointer, <RET> selects the option,  ?=help
```

■ **Figure 6-1. The Configuration Menu when the state and timing analyzers are both installed in the μAnalyst mainframe.**

00842-049

■ **Figure 6-2. Aligned/non-aligned acquisition modes.**

The following paragraphs describe the various aligned/non-aligned acquisition modes. Note that the mode descriptions appear in a different order than the modes appear on the screen. This is because the two aligned modes (State Arms Timing and Timing Arms State) contain background information that will be needed for the non-aligned mode (Crosslink Triggering).

State Arms Timing (Aligned)

In this mode, the state analyzer looks for an occurrence or sequence of occurrences before arming the timing analyzer. The Crosslink (CL) line is used to provide the arming signal. The word recognizers and triggering sequences in the state analyzer must be programmed so that CL will go low (0) when you want the timing analyzer to be armed.

Once arming occurs, the timing analyzer will be able to look for its own trigger pattern. If the timing analyzer's trigger pattern has been to set to all don't cares (Xs), it will trigger immediately.

There are some programming requirements in the State Arms Timing mode that must be followed. The requirements are listed below.

State Analyzer Programming Requirements, Aligned Mode, State Arms Timing (for information about the specific menus mentioned, see Section 5 of this manual):

1. Once the state analyzer asserts CL=0 in a trigger-store state, all of the following states must also assert CL=0. This way, CL remains asserted throughout the duration of the timing analyzer's acquisition.

2. All acquisition clocks must be stored, therefore:
    a. The Multiple-Preview Acquisition (MPA) mode is not available in the Trigger-Store Environment Submenu.

    b. The Store All State Transitions field must be set to YES in the Trigger-Store Environment Submenu.

    c. The storage qualifiers must all be set to ALWAYS STORE in the Trigger-Store Menu.

3. Only 14 programmable states are allowed in the Trigger-Store Menu. The fifteenth state is allocated for internal use.

Timing Analyzer Programming Requirements, Aligned Mode, State Arms Timing (for information about the specific menus mentioned, refer to the *Interactive Timing Analyzer Users Manual*):

1. The Multi-Triggers mode is not available in the Format Menu.

2. The timebase selection in the Format Menu is limited to 10 or 20 ns. This guarantees storage of the state analyzer's master clock pulse.

If the programming requirements for the state and timing analyzers have not been met, you will be prompted when attempting to leave the edit mode. You can let the system correct the programming automatically, or you can enter the menus and correct the programming manually, using the displayed error message as a guide.

Time-Aligned Display Requirements. In addition to the programming requirements listed earlier, there are certain conditions that must be met before data can be correlated for display:

1. The state and timing memories must both contain the sample event that caused the CL=0 assertion.

2. The data in the state and timing memories must overlap at some point in time (i.e., their time domains must intersect).

Position the state and timing triggers carefully to ensure that the above two conditions are met. Only the state and timing data that overlaps is correlated for display.

Timing Arms State (Aligned)

In this mode, an arming signal is generated when the timing analyzer's trigger condition is met. The Crosslink (CL) line is asserted low (0) and held until both the state and timing acquisitions are complete.

On the state analyzer side of this mode, the CL is monitored by a preprogrammed state. This state is internally programmed to look for CL going low. (This internally programmed state will not be seen in the state analyzer's Trigger-Store Menu.)

Once the CL is asserted, the state analyzer goes to the start state you programmed in the Trigger-Store Environment Submenu. If this start state is programmed to trigger on all don't cares, the state analyzer will trigger immediately. Otherwise, it will execute the state sequences as programmed.

As with the State Arms Timing mode, there are some programming requirements in the Timing Arms State mode that must be followed. These requirements are listed below.

State Analyzer Programming Requirements, Aligned Mode, Timing Arms State (for information about the specific menus mentioned, see Section 5 of this manual):

1. All the states defined in the Trigger-Store Menu must be set to CL=1. In order for the Timing Arms State mode to work properly, only the timing analyzer can be allowed to assert CL=0.

2. All acquisition clocks must be stored, therefore:

    a. The Multiple-Preview Acquisition (MPA) mode is not available in the Trigger-Store Environment Submenu.

    b. The Store All State Transitions field must be set to YES in the Trigger-Store Environment Submenu.

    c. The storage qualifiers must all be set to ALWAYS STORE in the Trigger-Store Menu.

3. Only 14 programmable states are allowed in the Trigger-Store Menu. The fifteenth state is used internally to look for the arming signal from the timing analyzer.

Timing Analyzer Programming Requirements, Aligned Mode, Timing Arms State (for information about the specific menus mentioned, refer to the *Interactive Timing Analyzer Users Manual*):

1. The Multi-Triggers Mode is not available in the Format Menu.

2. The timebase selection in the Format Menu is limited to 10 or 20 ns. This guarantees storage of the state analyzer's master clock pulse.

If the programming requirements for the state and timing analyzers have not been met, you will be prompted when attempting to leave the edit mode. You can let the system correct the programming automatically, or you can go back and change it manually, using the displayed error message as a guide.

Time-Aligned Display Requirements. In addition to the programming requirements listed earlier, there are certain conditions that must be met before data can be correlated for display:

1. The state and timing memories must both contain the sample event that caused the CL=0 assertion.

2. The data in the state and timing memories must overlap at some point in time (i.e., their time domains must intersect).

Position the state and timing triggers carefully to ensure that the above two conditions are met. Only the state and timing data that overlaps is correlated for display.

Crosslink Triggering (Non-Aligned)

In the non-aligned mode, the state and timing analyzers can arm each other, but their respective data is not time-aligned for the display. This allows greater flexibility in the available cross-arming and triggering setups. You can use any of the triggering modes, including Multiple-Preview Acquisition in the state analyzer, and Multi-Triggers in the timing analyzer. In addition, you can use any of the timing analyzer's timebase selections.

Note that, as with any trigger mode, it is possible to program arming/triggering sequences in which the state or timing triggers may never occur. Think carefully about your trigger and crosslink programming in both analyzers.

In the non-aligned mode, the state and timing analyzers can both sense and/or assert the Crosslink (CL) line. In the state analyzer, the CL is sensed within each defined symbol in the Symbol Menu, and the CL is asserted within each defined state in the Trigger-Store Menu.
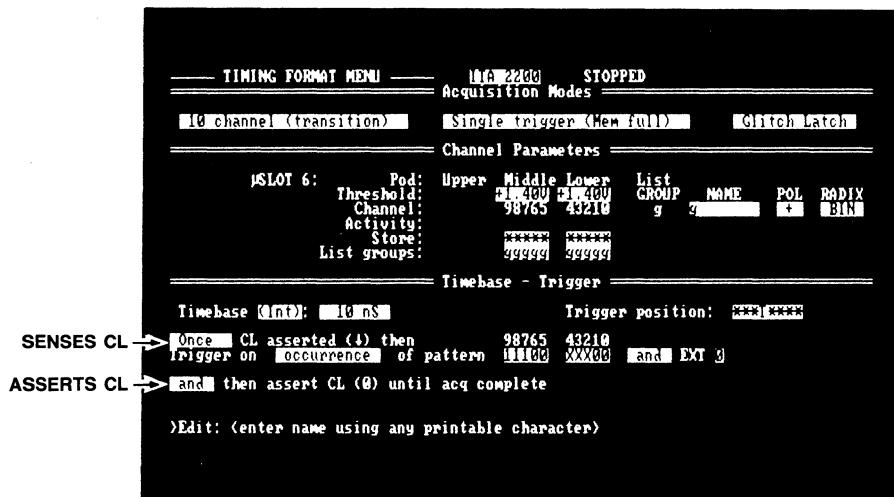
In the timing analyzer, two extra CL programming fields appear in the Format Menu. As shown in Figure 6-3, one field appears above the trigger pattern and the other appears below the trigger pattern.

The field which appears above the pattern is used for sensing the CL signal asserted by the state analyzer. Using this field, you can specify that the timing analyzer look for its trigger once the state analyzer has asserted CL, or you can specify that timing analyzer looks for its trigger while the state analyzer is asserting CL (thus, creating a trigger window).

The field which appears below the pattern is used for asserting CL from the timing analyzer.

00842-050

```
—— TIMING FORMAT MENU ——    ▌TA 2200▐    STOPPED
                           Acquisition Modes
  ▌10 channel (transition)▐   ▌Single trigger (Mem full)▐   ▌Glitch Latch▐
                           Channel Parameters
        ▐SLOT 6:        Pod:  Upper Middle Lower   List
                  Threshold:  ▌1.40U▐ ▌1.40U▐       GROUP   NAME   POL  RADIX
                    Channel:  98765 43210            g    ▌?        ▐  ▌+▐  ▌BIN▐
                   Activity:
                      Store:  ▒▒▒▒▒ ▒▒▒▒▒
                List groups:  ggggg ggggg
                           Timebase - Trigger
      Timebase ▌Int▐:  ▌10 nS▐              Trigger position:  ▒▒▒▌▒▒▒▒
SENSES CL ➝ ▌Once▐ CL asserted (↓) then         98765 43210
         Trigger on ▌occurrence▐ of pattern ▌11100▐ XXX00 ▌and▐ EXT ▌0▐
ASSERTS CL ➝ ▌and▐ then assert CL (0) until acq complete

         )Edit: (enter name using any printable character)
```

■ **Figure 6-3. The Timing Analyzer's Format Menu when the nonaligned mode is selected.**

As a simple example of the flexibility of the non-aligned mode, the timing analyzer could (1), be armed by the state analyzer; (2), once armed, it could trigger on its own programmed trigger pattern; and then (3), upon triggering, it could assert the CL to arm or trigger the state analyzer.

████ USING STATE & TIMING TOGETHER

You enter the aligned/non-aligned menus differently than you enter the individual state and timing menu systems. The following paragraphs describe how you enter the modes, then the menus.

Entering the Aligned/Non-Aligned Modes

The aligned/non-aligned modes can be entered at any time. To do so, first enter the Configuration Menu (press M for Menus and C for Configuration) and select the State & Timing Analyzers option. The Configuration Menu will then list the aligned/non-aligned mode choices.

Select your mode choice by pressing the return key. If your choice is one of the aligned modes and if the current state sequence or programming does not meet the aligned mode requirements dis-cussed earlier, you will be prompted with the following message:

"Make all corrections to parameter violations (Y/N)?"

If you press Y for Yes, the μAnalyst will make all the necessary programming corrections. If you press N for No, you will need to select another choice.

Once you have made a mode choice, press the return key again. The μAnalyst will enter the state and timing menus.

Operating the State & Timing Menus

When using the state and timing analyzers together, or when observing their correlated acquisitions, you may need to jump back and forth between the state and timing analyzer menu systems and command lines.

To do this, press M for Menus from the command line. The current set of analyzer menus available (state or timing) is indicated on the command line, in brackets, in bold video. Press function key F1 to select STATE menus or press function key F2 to select TIMING menus. The menu list will then change accordingly.
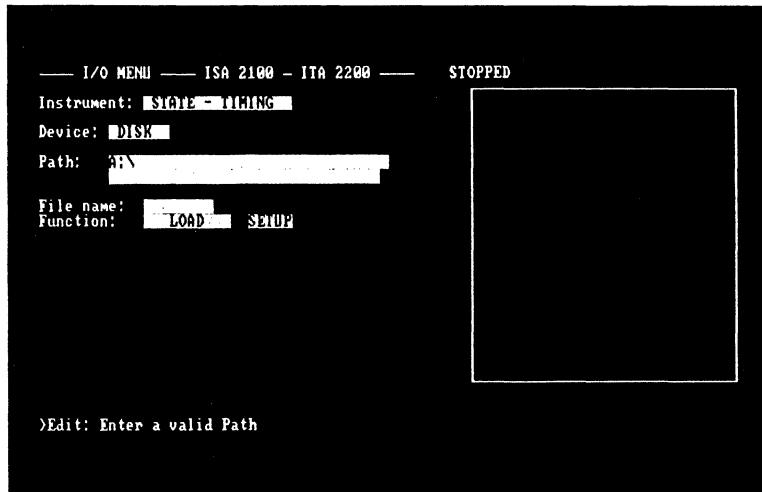
If at some point you need to identify which set of menus you are using, simply press M for Menus. You'll see which menus are cur-rently active. Once familiar with both analyzers, you will notice that the state and timing menus are quite different and easy to distinguish.

Saving Files in the I/O Menu

If you are operating the state and timing analyzers in the non-aligned mode, the I/O Menu provides an additional field, labeled Instrument. As shown in Figure 6-4, this field lets you specify which instrument is used during the I/O operation. You can select STATE, TIMING, or STATE & TIMING.

If you are operating the state and timing analyzers in an aligned mode, the Instrument field is not programmable. It is automatically set to STATE & TIMING, so that the I/O operation is performed on both analyzers.

00842-051

```
—— I/O MENU —— ISA 2100 – ITA 2200 ——    STOPPED
Instrument: STATE – TIMING
Device: DISK
Path:   A:\

File name:
Function:      LOAD    SETUP




)Edit: Enter a valid Path
```

■ **Figure 6-4. The I/O Menu, when the state and timing non-aligned mode is selected.**

████ SPLIT-SCREEN DISPLAY

A split-screen display mode is available if either of the aligned modes has been selected. As shown in Figure 6-5, this mode allows the simultaneous display of both timing and state data.
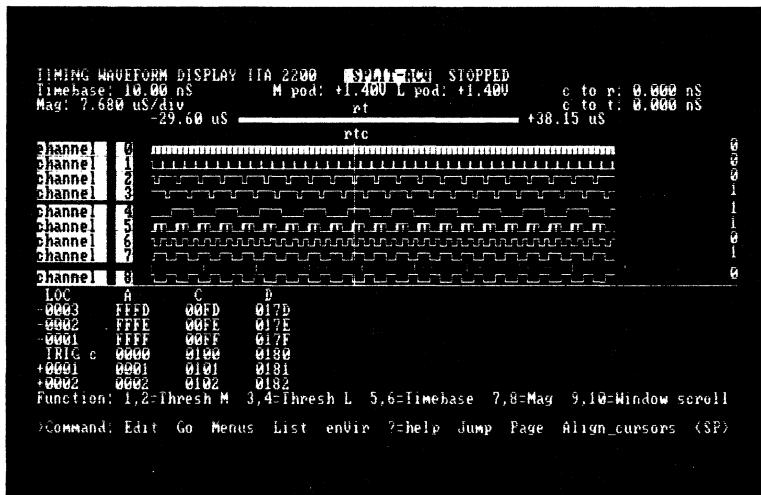
☑ **NOTE**

It is assumed that the acquired state and timing data falls within the same time domain, or that at the least the domains intersect.

In the split-screen display, both sets of data are shown in their normal display mode. That is, timing data is shown in a waveform display, and state data is shown in a columnar radix display.

The split-screen shows up to 20 data lines, divided into two windows. You can specify the relative size of the windows in the timing analyzer's Environment Submenu. As shown in Figure 6-6, the submenu provides a field labeled State Split-Screen Size. This field determines how many lines of the display are dedicated to the state analyzer. You can enter any value into the field between 1 and 10, inclusive.

00842-052

■ **Figure 6-5. Split-screen display.**

00842-053

```
TIMING WAVEFORM DISPLAY ITA 2200   █SPLIT-ACQ STOPPED
Timebase: 10.00 nS        M pod: +1.40V L pod: +1.40V      c to r: 0.000 nS
Mag: 7.680 uS/div                      rt               c to t: 0.000 nS
         -29.60 uS ─────────────────────── +38.15 uS
                               rtc
 channel ▌0 ▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄                    0
 channel ▌1 ▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄                    0
 channel ▌2 ▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄                    0
 channel ▌3 ▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄                    1
 channel ▌4 ▄▄  ▄▄  ▄▄  ▄▄  ▄▄  ▄▄  ▄▄  ▄▄  ▄▄                       1

════════════════════════ TIMING Environment Submenu ════════════════════════
Acquisition mode:     █ GO ONCE █      Compare ACQ to REF (highlight): █NO█
                                                 From: -1.000█  mS
                                                   To: +1.000█  mS
 List display mode: █ STANDARD █       Acq offset (+/-): 0   locations
                                          Fuzz (+/-): 0   samples
 List time readout: █RELATIVE█                         FEDCBA9876543210
                                          Using mask of: ****************
 List page increment size: 30
                                        Non-overlap range: █ COMPARES EQUAL █
→ State split-screen size: 36

 Message: ESC returns to TIMING LIST DISPLAY
 >Edit: SPACE scrolls through choices  ?=help
```

■ **Figure 6-6. Selecting the size of the split-screen display in the timing analyzer's Environment Submenu.**

All the display controls that are normally available for state and timing displays, are available for the split-screen display. The scrolling of the two display windows works in a master-slave fashion. For example, if you use the up and down cursor keys to scroll the state data, the timing waveforms are automatically scrolled left-to-right to the next corresponding state clock region.

If you use the left or right cursor keys to scroll the timing waveforms, the waveform display becomes the master. Calculations are made by the software to determine if the waveform scrolling has moved into a new state clock region. If so, the state display is scrolled accordingly. Should the scrolling action of one display force the other past its data, a message will advise you that the system has been scrolled beyond time-aligned data.

The split-screen mode becomes the default display mode when an aligned mode is selected. To change to another display mode, go to the waveform portion of the display, enter the edit mode, and select another choice in the Memory Selection field.
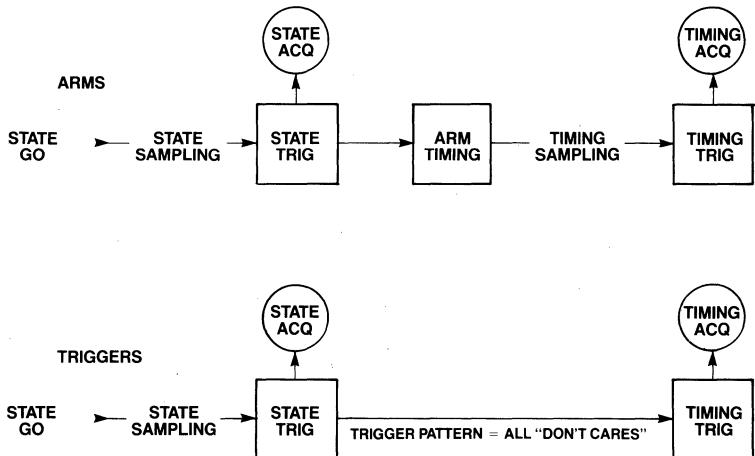
▆▆▆▆▆▆ TYPICAL STATE & TIMING APPLICATIONS
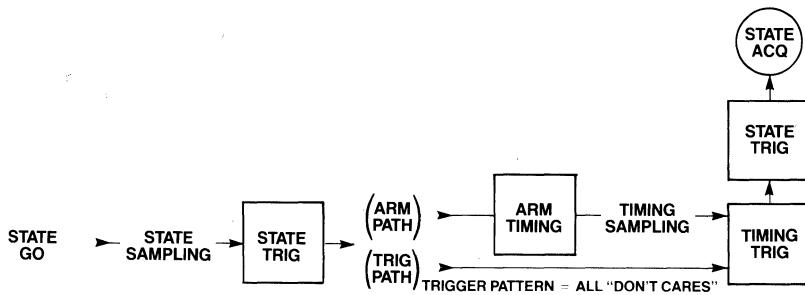
Figures 6-7 and 6-8 illustrate some of the more common ways in which the two analyzers can be combined to arm/trigger one another.

■ **Figure 6-7. State Arms/Triggers Timing.**

■ **Figure 6-8. State Arms/Triggers Timing, Timing Triggers State.**

Using this section. This section gives you an opportunity to operate the Interactive State Analyzer before getting into actual applications. The demonstration example is built around the demo circuit board, which you received as part of your state analyzer package.

Before attempting the demonstration example, it is recommended that you first read Sections 2, 3 and 4 of this manual. These sections show you how to install the μAnalyst hardware and software, and they provide important background information on the state analyzer features.

SECTION CONTENTS

■■■ SYSTEM SETUP

The μAnalyst products and accessories you'll need to set up the demo system are listed below:

- μAnalyst 2000 Mainframe, with the personal computer interface boards and cable.

- State Analyzer Controller Board and one P2901 Clock Probe

- State Memory Board (either 16- or 32-channel) and one P2902 Data Probe.

- μAnalyst software (SYSTEM DISK and HELP DISK).

- Demo Circuit Board

In addition to the above μAnalyst accessories, you'll also need a personal computer that meets the PC Compatibility Requirements listed at the front of this manual.

Installing the μAnalyst and the Personal Computer. If you haven't already done so, install the PC interface board into your personal computer. Also install the μAnalyst interface board and cable, the controller board, and the memory boards as described in Section 2 of this manual.

Connecting the Probes. Now, connect the clock and data probes as follows:

1. Attach the P2901 Clock Probe to the state analyzer controller board.

2. Attach the P2902 Data Probe to the state memory board that is farthest from the controller board. If this farthest board is a 32-channel memory board, attach the probe to the board's upper probe connector.

☞ **NOTE**
It is important that you connect the probes exactly as described above. Otherwise, the demonstration example will not work properly.
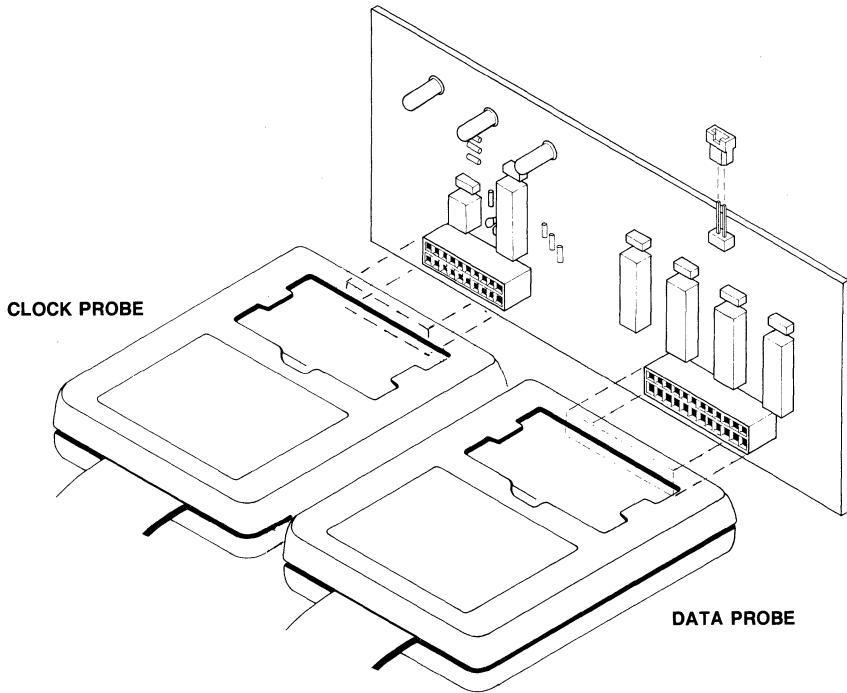
Installing the Demo Board. To connect the demo circuit board to the μAnalyst, follow these steps, referring to Figure 7-1:

1. Connect the P2901 Clock Probe (from the Controller Board) to Clock Probe connector on the demo board.

2. Connect the P2902 Data Probe (from the Memory Board) to the Data Probe connector on the demo board.

When connecting a probe to the demo board, it is easiest if you hold the demo board vertically so that the two block connectors are at the bottom and horizontal. Center the probe's shrouded pin header over the board's connector, then push the probe into contact with the demo board.

00842-055

CLOCK PROBE

DATA PROBE

■ **Figure 7-1. Installing the demo circuit board.**

Starting the System. To start the system, perform these steps:

1. Install a DOS diskette (version 2.00 or higher) into drive A of your personal computer.

2. Power up both the μAnalyst mainframe and the personal computer.

3. Remove the DOS diskette from Drive A and replace it with the μAnalyst's HELP DISK.

4. Insert the μAnalyst's SYSTEM DISK into drive B of the personal computer.

5. Enter drive B, then type the word ANALYZE and press the return key.

In a few moments you'll see the Configuration Menu on the screen. You're ready to start the demonstration.

☞ **NOTE**

If you're using a hard-disk computer, load the μAnalyst software as specified in Section 3, *Getting Started*.

■■■ THE CONFIGURATION MENU

If you've followed the instructions in the preceding subsection, you should now be looking at the Configuration Menu. Your screen should be similar to the one shown in Figure 7-2, although there may be some differences in the number and depth of memory cards, and perhaps the location of the controller and memory cards.
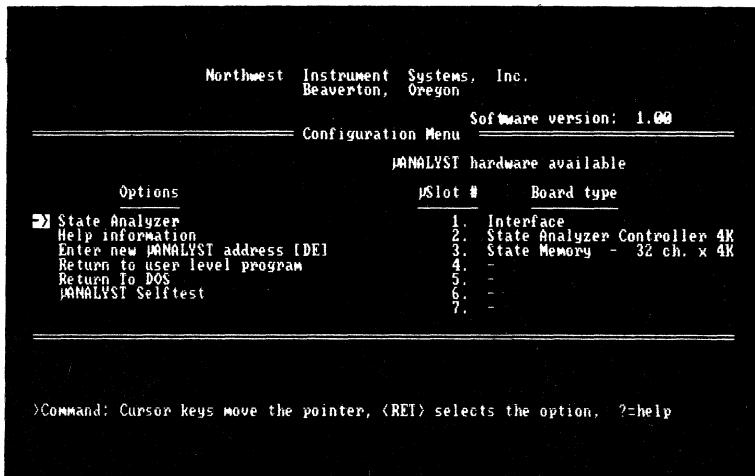
If your screen looks like the one shown in Figure 7-3, one of three things is probably wrong:

1. The power to the μAnalyst is not on.

2. The cable between the computer and the μAnalyst is not connected.

3. The PC interface board (the small board that plugs into the personal computer I/O bus) is set to a different address range than the address range shown on the Configuration Menu.

00842-056

```
                  Northwest  Instrument  Systems,  Inc.
                             Beaverton,  Oregon
                                           Software version:  1.00
            ==================== Configuration Menu ====================

                              µANALYST hardware available

            Options                       µSlot #      Board type
            _____                      _____       _____
        ▶▶ State Analyzer                    1.     Interface
           Help information                  2.     State Analyzer Controller 4K
           Enter new µANALYST address [DE]   3.     State Memory  -  32 ch. x 4K
           Return to user level program      4.     -
           Return To DOS                     5.     -
           µANALYST Selftest                 6.     -
                                             7.     -



        >Command: Cursor keys move the pointer, <RET> selects the option,  ?=help
```
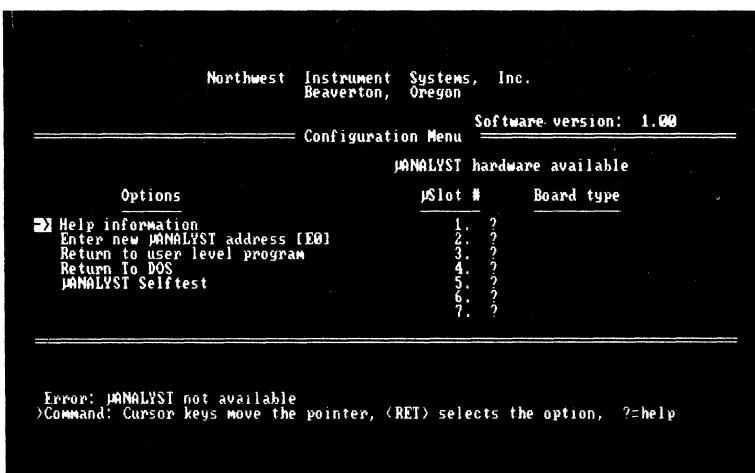
■ **Figure 7-2. The Configuration Menu.**

00842-057

```
                   Northwest  Instrument  Systems,  Inc.
                              Beaverton,  Oregon
                                            Software version:  1.00
             ==================== Configuration Menu ====================

                               µANALYST hardware available

             Options                       µSlot #      Board type
             _____                      _____       _____
         ▶▶ Help information                  1.    ?
            Enter new µANALYST address [E0]   2.    ?
            Return to user level program      3.    ?
            Return To DOS                     4.    ?
            µANALYST Selftest                 5.    ?
                                              6.    ?
                                              7.    ?



         Error: µANALYST not available
         >Command: Cursor keys move the pointer, <RET> selects the option,  ?=help
```

■ **Figure 7-3. The Configuration Menu, without proper connections.**

Take appropriate action to correct problems (1) or (2). If the problem is (3), first determine the address programmed on your PC interface board. Now, using the down cursor key, move the screen pointer to the menu's Enter New μAnalyst Address option, as shown in Figure 7-4. Press the return key. The address field in the option should immediately start to blink. In the field, enter the two hexadecimal characters which correspond to the eight most significant bits of the address programmed on the PC interface board.

☑ **NOTE**

If the above procedures fail to take care of the problem, refer to Section 3 of this manual for instructions on how to handle power-up error conditions.

00842-058

```
                    Northwest  Instrument  Systems,  Inc.
                    Beaverton,  Oregon
                                          Software version:  1.00
    ═══════════════════════ Configuration Menu ═══════════════════

                                          μANALYST hardware available

              Options                     μSlot #      Board type

         State Analyzer                     1.   Interface
         Help information                   2.   State Analyzer Controller 4K
    ▶   Enter new μANALYST address [00]     3.   State Memory  -  32 ch. x 4K
         Return to user level program       4.   -
         Return To DOS                       5.   -
         μANALYST Selftest                   6.   -
                                             7.   -



    ═══════════════════════════════════════════════════════════════


     Message: Enter the 8 ms bits (in HEX) for μANALYST interface address,  ?=help
     >Edit: Enter 0 thru F  ?=help
```

■ **Figure 7-4. Changing the PC interface board's address in the Configuration Menu.**

Now that your system is up and running, take a few minutes to read the on-line Help information.

This Help file gives basic information about how to move the cursor through the menus, select commands, and enter and leave the edit mode within a menu.

Help also explains the command and message lines at the bottom of the screen. These lines are very important in using the system. The command line provides supplemental information about the command you have selected or, if you've made a mistake, gives you an error message. Any time you are unsure of what to do next, read the command and message lines. In most cases this will provide you with the information you need to proceed.

To read the Help information, move the screen pointer to the Help Information option of the Configuration Menu and press the return key. The bottom line of the screen will now show three commands:

< PgUp >   Press this key to move backward through the Help file.

< PgDn >   Press this key to move forward through the Help file.

< Esc >   Press this key to escape the Help file and return to the Configuration Menu.

Now that you have read the Help information, we can proceed with the demonstration. In this demonstration we will be monitoring and controlling the demo board that you have connected to the clock and data probes.

The demo board consists of a 16-bit up/down counter. The state analyzer can be programmed to control the up/down function of the counter, its load initial count function, and its counter enable input. Of course, the state analyzer also monitors the counter's 16 output lines.

As we proceed through the demonstration, we will discuss the operation of the demo board in more detail, and we'll clarify exactly how to program the state analyzer to control and/or monitor the board.

First let's go to the I/O Menu and load a setup file, which will automatically program the entire operation of the state analyzer. Move the screen pointer to the State Analyzer option in the Configuration Menu, and press the return key.

The state analyzer will display its Format Menu first. To move from the Format Menu to the I/O Menu, use the Menu command on the command line at the bottom of the screen. Press M. The command line now lists all of the available menus, including the I/O Menu. Press I to select the I/O Menu.

▪▪▪ THE I/O MENU

In the I/O Menu, you should see a screen like the one in Figure 7-5.

There are five programmable fields in this menu. The Device field allows you to select either disk or printer as the I/O device. Disk is the currently selected device.

00842-059
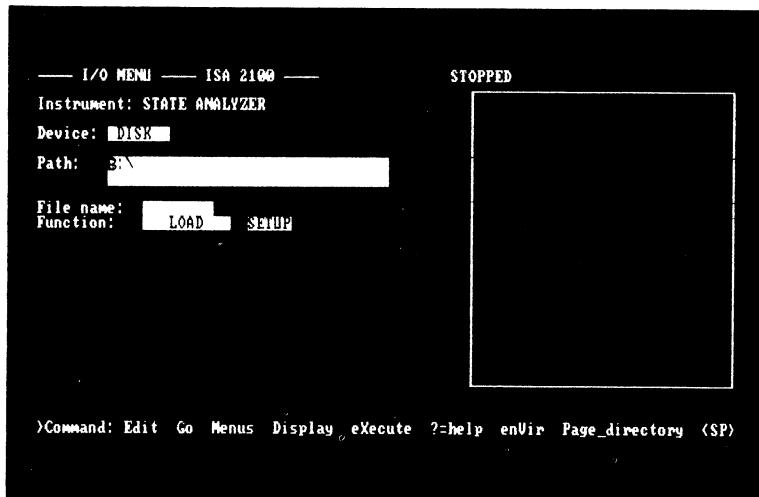
■ **Figure 7-5. The I/O Menu.**

The Path field refers to the current drive-directory in use. The state analyzer files contained at the path location are displayed in the directory window on the right-hand side of the screen.

The File Name field is used to indicate the name of the file you wish to act on.

The Function field allows you to describe what you want to do with the indicated file.

In this demonstration we want to use the I/O Menu to load the State Analyzer Demo Program, which is contained on the μAnalyst's HELP DISK. The following steps show you how to do this.

1. Press E to enter the edit mode.

2. Move the screen cursor to the Path field and enter:

   **A: **

   Now press the return key. The directory window should immediately show you the files on the HELP DISK.

3. Move the screen cursor to the File Name field, then enter the name of the DEMOXX file that matches the total number of channels in your state analyzer. For example, if your state analyzer has a total of 32 channels, enter DEMO32.

4. Move the screen cursor to the Function field, then, using the space bar, set the field to:

   LOAD SETUP

5. Finally, press the escape key to leave the edit mode and return to the command mode.
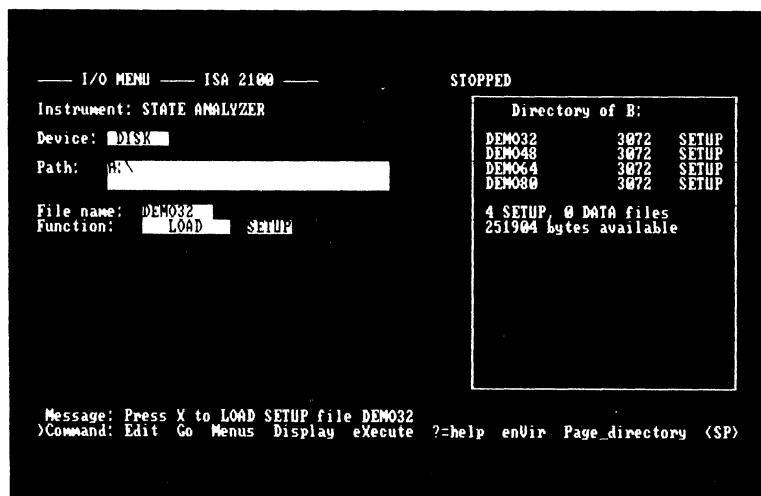
Your I/O Menu should now look similar to the one shown in Figure 7-6.

☑ **NOTE**

In this example, we used the DEMO32 file because our state analyzer is configured with a total of 32 channels. If your state analyzer contains more channels, you should be using the appropriate file. Otherwise, the demonstration example will not work properly.

00842-060

```
 ── I/O MENU ── ISA 2100 ──              STOPPED
Instrument: STATE ANALYZER                  Directory of B:
Device: ▮DISK▮                          DEMO32        3072    SETUP
Path:    B:\                            DEMO48        3072    SETUP
                                        DEMO64        3072    SETUP
                                        DEMO80        3072    SETUP
File name:  DEMO32
Function:      LOAD    SETUP            4 SETUP, 0 DATA files
                                        251904 bytes available




 Message: Press X to LOAD SETUP file DEMO32
 >Command: Edit  Go  Menus  Display  eXecute  ?=help  enVir  Page_directory  <SP>
```

■ **Figure 7-6. Setting up the I/O Menu to load the demo file.**
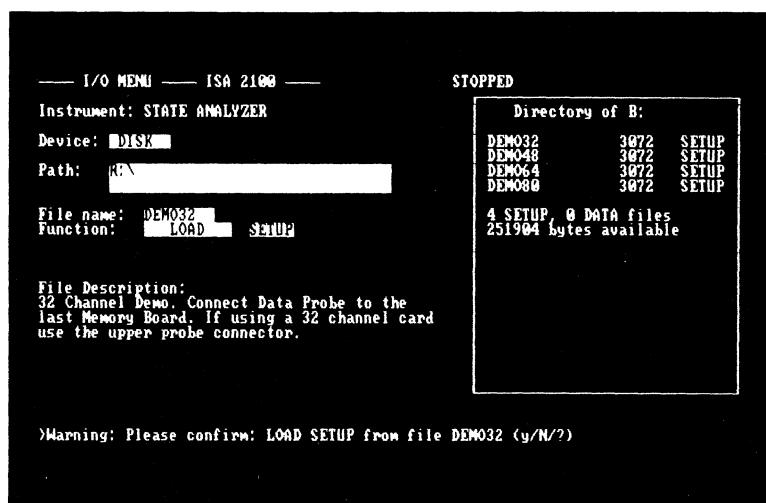
You can now load the demo file by pressing X for eXecute. Your screen should now look like Figure 7-7.

The file description reminds you that the data probe connected to the demo board must be attached to the memory board farthest from the controller board, and it tells you how many channels this demo file is set up for.
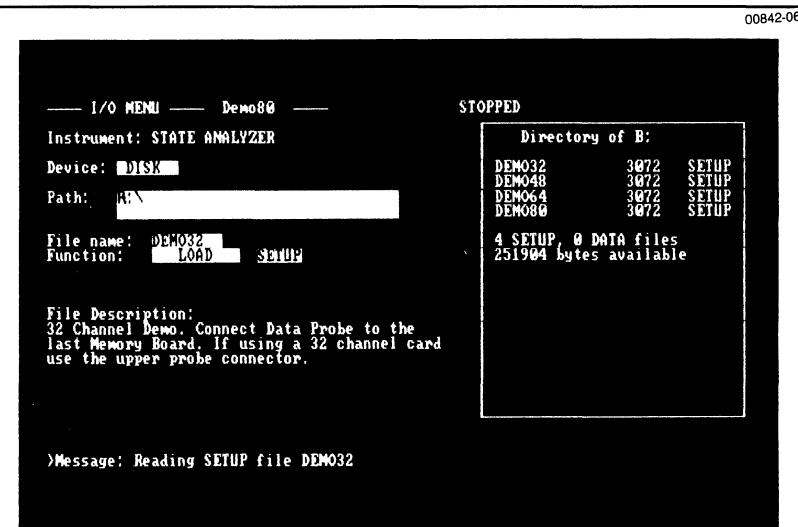
If the number of channels in the file description matches the number in your system, enter Y for Yes when the μAnalyst asks you to confirm the loading operation.

00842-061

```
—— I/O MENU —— ISA 2100 ——        STOPPED

Instrument: STATE ANALYZER
                                  ┌────────────────────────────────┐
Device: ▮DISK▮                    │   Directory of B:              │
                                  │                                │
Path:   R:\�_____       │ DEMO32        3072   SETUP      │
        ▮_____  │ DEMO48        3072   SETUP      │
                                  │ DEMO64        3072   SETUP      │
File name:  DEMO32▮                │ DEMO80        3072   SETUP      │
Function:   ▮LOAD▮  SETUP          │                                │
                                  │ 4 SETUP, 0 DATA files          │
                                  │ 251904 bytes available         │
File Description:                  │                                │
32 Channel Demo. Connect Data Probe to the    │                    │
last Memory Board. If using a 32 channel card │                    │
use the upper probe connector.    │                                │
                                  │                                │
                                  └────────────────────────────────┘


>Warning: Please confirm: LOAD SETUP from file DEMO32 (y/N/?)
```

■ **Figure 7-7. Loading the demo file.**

During the few moments the setup is loading, your screen will look like Figure 7-8.

00842-062

```
 ──── I/O MENU ──── Demo80 ────        STOPPED

Instrument: STATE ANALYZER                ┌─────────────────────────────┐
                                          │     Directory of B:         │
Device: ▮DISK▮                            │                             │
                                          │ DEMO32       3072    SETUP   │
Path:  B:\                                │ DEMO48       3072    SETUP   │
                                          │ DEMO64       3072    SETUP   │
File name:  DEMO32                        │ DEMO80       3072    SETUP   │
Function:      LOAD    SETUP              │                             │
                                          │ 4 SETUP, 0 DATA files       │
                                          │ 251904 bytes available      │
File Description:                         │                             │
32 Channel Demo. Connect Data Probe to the│                             │
last Memory Board. If using a 32 channel card│                          │
use the upper probe connector.            │                             │
                                          │                             │
                                          │                             │
                                          └─────────────────────────────┘

>Message: Reading SETUP file DEMO32
```

■ Figure 7-8. Reading the setup file.
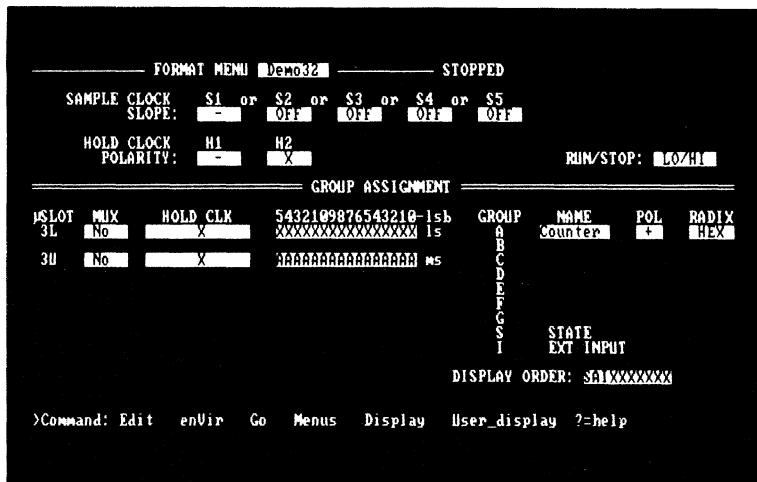
When the command line reappears, move to the Format Menu: first select the Menu command by pressing M, and then select the Format Menu by pressing F.

7-13

Your screen should now look like Figure 7-9.

00842-063

■ **Figure 7-9. The Format Menu.**

▬▬▬ THE FORMAT MENU

The Format menu is used to describe how you have con-
nected the state analyzer to the device under test, and how this set
of connections should be grouped, named and displayed. It also
allows you to name the setup (just as we named the demo setup
DEMOXX).

The demo board counter outputs are connected to the chan-
nels marked A in Figure 7-9 and on your screen. For the configura-
tion shown in Figure 7-9, you can see that the data probe connected
to the demo board is received by the upper probe connector of the
32-Channel Memory Board in slot number 3.

This group of lines has been named Counter on the right-
hand side of the screen under the NAME column. They are to be
displayed in positive logic (POL=+) and in hexadecimal notation
(RADIX=HEX). In the Display Menu, the data acquired from these
lines will be in a column that is titled Counter.

We're going to defer further explanation of the Format Menu
until we've run the state analyzer a few times and have learned a little
more about how the demo is constructed.
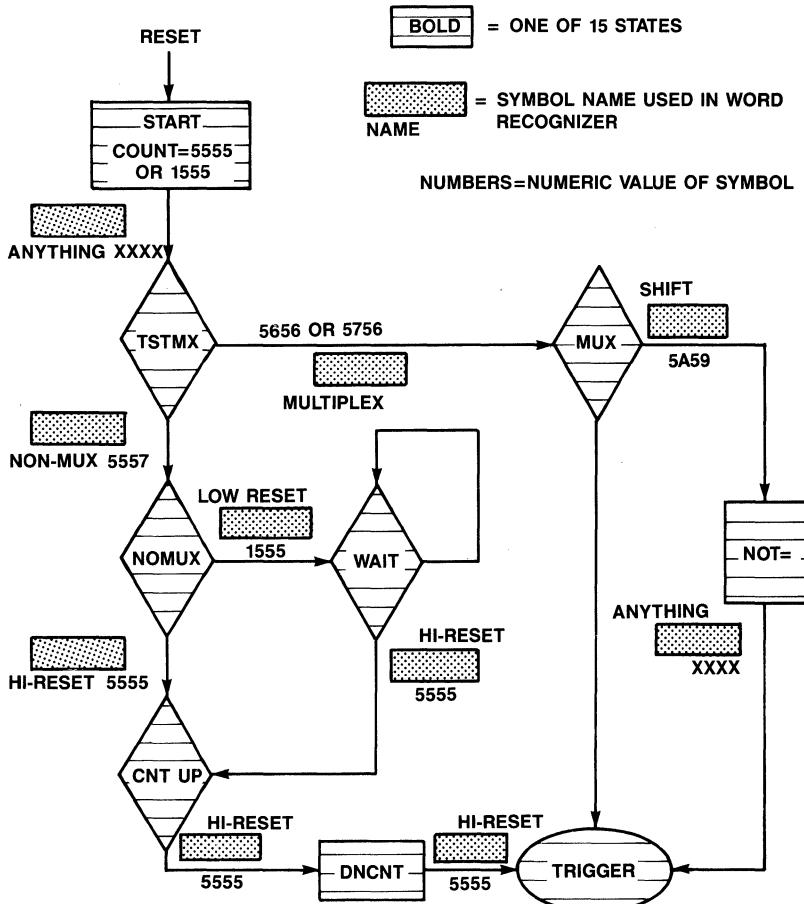
The *Trigger-Trace Flowchart* in Figure 7-10 is a flowchart of the trigger and trace program that was loaded into the demo with the DEMOXX setup file. It is a graphical representation of how the triggering is currently programmed. We will get to know how this is done a little later.

For now, notice that there are three different paths to the TRIGGER oval in the lower right-hand corner of the chart. We will cause the data presented to the state analyzer to be different each time we run the machine, by making changes in the Format Menu. (How, we will explain later. For now, our object is to learn to start the state analyzer, follow the flowchart, and manipulate the Format Menu.)
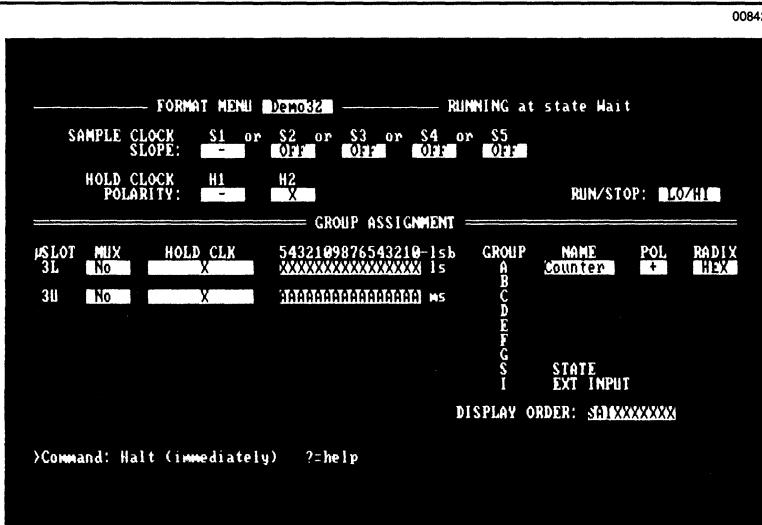
00842-064

■ Figure 7-10. Demonstration Setup. Trigger-Trace Flowchart.

Let's get started. Connect the jumper on the demo board to both pins. Select the Go command by pressing G. Your screen should now look like Figure 7-11.

00842-065

■ **Figure 7-11. Running at state WAIT.**

The message in the upper right-hand corner tells you that the state analyzer is running at the state WAIT. Find the box labeled WAIT on your flowchart. The trigger sequence is stuck in state WAIT and is waiting for you to remove the jumper from the demo board. Do this now while you watch the "RUNNING..." message. You probably noticed the "RUNNING..." message indicate states CNTUP and DNCNT before changing to "STOPPED at state DNCNT."
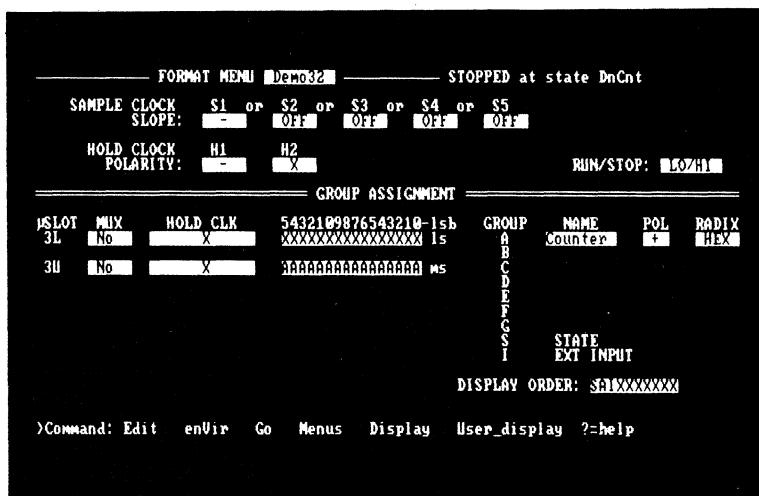
(Leave the jumper off for the time being. You might want to attach it to one leg only on the demo board. That's fine, but be sure it's only on one leg, or you will see slightly different displays from the ones we'll be showing you.)

Your screen should now look like the one shown in Figure 7-12.

```
──────────── FORMAT MENU ▮Demo32▮ ──────────── STOPPED at state DnCnt

    SAMPLE CLOCK    S1   or   S2   or   S3   or   S4   or   S5
          SLOPE:  ▮  -  ▮   ▮OFF▮   ▮OFF▮   ▮OFF▮   ▮OFF▮

    HOLD CLOCK    H1        H2
       POLARITY: ▮  -  ▮   ▮  X  ▮                  RUN/STOP: ▮LOZHI▮
═════════════════════════ GROUP ASSIGNMENT ═════════════════════

 µSLOT  MUX    HOLD CLK     543210987654321 0-lsb  GROUP    NAME    POL  RADIX
  3L  ▮No▮  ▮      X      ▮ XXXXXXXXXXXXXXXX 1s      A    ▮counter▮ ▮+▮ ▮HEX▮
                                                    B
  3U  ▮No▮  ▮      X      ▮ }AAAAAAAAAAAAAAA ms      C
                                                    D
                                                    E
                                                    F
                                                    G
                                                    S    STATE
                                                    I    EXT INPUT

                                    DISPLAY ORDER: ▮SAIXXXXXXX▮


  )Command: Edit   enVir   Go   Menus   Display   User_display   ?=help
```

■ **Figure 7-12. Trigger stopped at state DNCNT.**
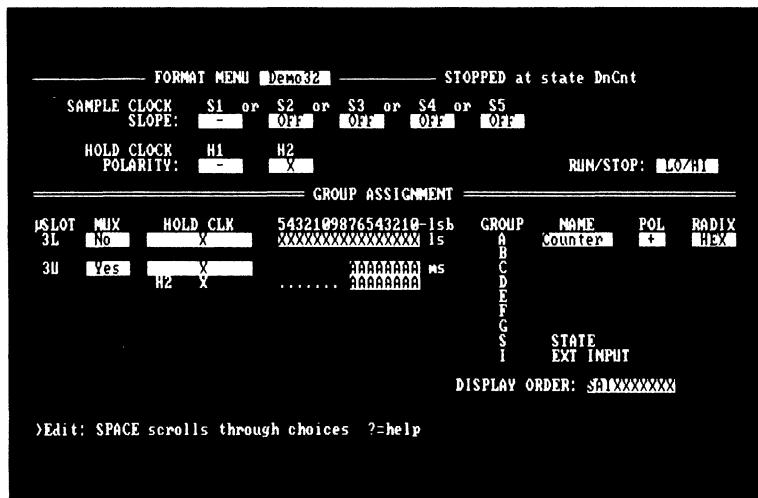
Now let's cause the triggering sequence to trigger in the mul-
tiplexed state (state MUX). To do this you must alter the Format Menu
programming. First enter the edit mode by pressing E for Edit.

If the blinking screen cursor is not over the MUX field in the row containing the A channels, move it there now. Scroll through your choices in the MUX field by pressing the space bar, as directed on the edit line. Select YES. Your screen should now look like Figure 7-13.

☞ **NOTE**

If your data probe is not connected to the memory board farthest from the controller board, this MUX operation will not perform as described above. Go back to the beginning of this section and reconnect your data probe as specified. Then, use the I/O Menu to load the DEMOXX file that fits your channel configuration.

00842-067
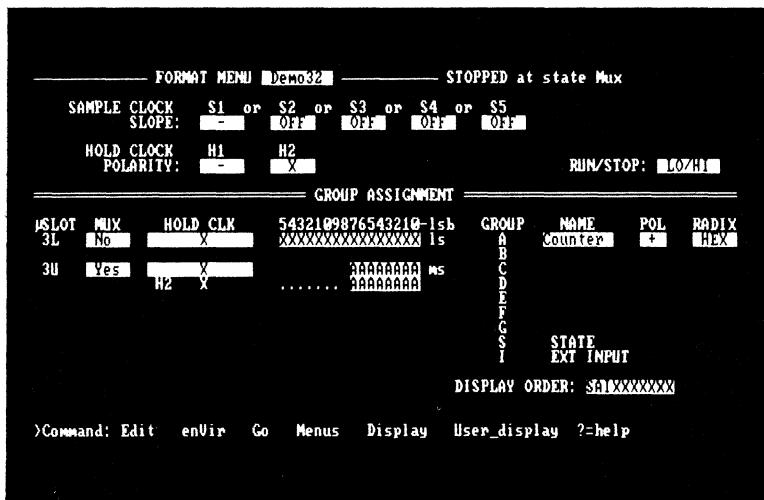
■ **Figure 7-13. Setting up multiplexed mode.**

Escape from the edit mode and select the Go command by pressing G. Your screen should look like Figure 7-14.

00842-068

```
———————————— FORMAT MENU █Demo32█ ———————— STOPPED at state Mux
      SAMPLE CLOCK    S1  or  S2  or  S3  or  S4  or  S5
            SLOPE:   █ - █   █OFF█   █OFF█   █OFF█   █OFF█

      HOLD CLOCK   H1      H2
         POLARITY: █ - █   █ X █                         RUN/STOP: █LO/HI█
      ════════════════════════ GROUP ASSIGNMENT ═════════════════
  ↓SLOT  MUX      HOLD CLK     5432109876543210-lsb  GROUP   NAME      POL   RADIX
    3L  █ No █   █      X    █  XXXXXXXXXXXXXXXXX 1s     A    █Counter█  █ + █  █HEX █
                                                        B
    3U  █ Yes █  █      X    █   █AAAAAAA█ MS            C
              █H2█   X      █  ....... █AAAAAAA█          D
                                                        E
                                                        F
                                                        G
                                                        S    STATE
                                                        I    EXT INPUT
                                         DISPLAY ORDER: █SAIXXXXXXX█

  >Command: Edit   enVir   Go   Menus   Display   User_display   ?=help
```

■ **Figure 7-14. Go in multiplexed mode.**

As you can see, we triggered in state MUX, the second path in the flowchart. What has happened here is that the triggering specification has been programmed to detect whether or not data is being treated as multiplexed (MUX=Yes) or non-multiplexed (MUX=No). As you might suspect, this determination was made in state TESTMX on your flowchart.

For the final path in the flowchart, we are going to detect whether or not a hold clock is being used. Enter edit mode. Move the cursor to the Hold Clock column and use the space bar to select H1 LO=PASS. Escape from edit mode. Your screen should look like Figure 7-15.

00842-069

————————— FORMAT MENU Demo32 ————————— STOPPED at state Mux

SAMPLE CLOCK   S1 or S2 or S3 or S4 or S5
        SLOPE: ─   OFF   OFF   OFF   OFF

HOLD CLOCK   H1   H2
   POLARITY: ─   X   RUN/STOP: LO/HI

========================= GROUP ASSIGNMENT =========================

SLOT MUX   HOLD CLK   5432109876543210-lsb GROUP   NAME   POL RADIX
3L   No   X   XXXXXXXXXXXXXXXX ls   A   Counter   +   HEX
                                                B
3U   Yes   H1 LO=PASS   AAAAAAA ms   C
            H2   X   .... AAAAAAA   D
                                                E
                                                F
                                                G
                                                S   STATE
                                                I   EXT INPUT

                                    DISPLAY ORDER: SAIXXXXXXX

>Command: Edit   enVir   Go   Menus   Display   User_display   ?=help

■ **Figure 7-15. Setting up to detect a hold clock.**

7-21

Now select the Go command. The analyzer will have stopped in the state NOT= and your screen should look like Figure 7-16.

■ **Figure 7-16. Detecting a hold clock.**

Before you do anything else, reprogram the Format Menu so that the MUX field is set to NO and the Hold Clock field is X. Your screen should look like Figure 7-9, with a "STOPPED..." message added.

If you are willing to take it on faith that we changed the data by altering the Format Menu, skip ahead a few pages to the material on the Trigger-Store Menu. Whether you do or not, it would be a good idea to read the material on sample and hold clocking provided in Section 4, *Modes of Operation*.

■ BUS DEMULTIPLEXING

The way you program the Format Menu affects the way data is acquired and displayed. This subsection discusses how the sample and hold clocks work together to capture different kinds of data.

As explained in the material on sample and hold clocking in Section 4, the state analyzer in MUX mode (where MUX=YES) looks at only the lower 8 lines of the data probe. These lines are run through two transparent (or presample) latches in parallel, so that the 8 lines are duplicated and two sets of 8 are presented to the rest of the memory board (shown on paths A and B in Figure 7-17.) Each of these sets of 8 can be presampled with its own hold clock and held until the data is clocked into the state analyzer with the S1-S5 sample clocks.

If both H1 and H2 are high and MUX=YES, then both 8-bit paths will present the same data to the internal logic of the state analyzer's memory board. However, if one or both of the hold clocks are active, and if the phase relationship between the hold clocks is not the same, then different sets of data can be clocked into the internal logic of the memory board.

00842-071
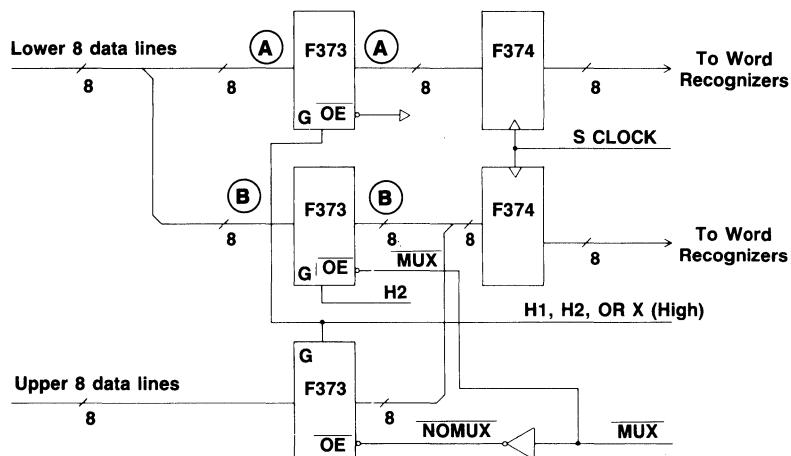
■ **Figure 7-17. Input circuitry of the memory board.**

Look at Figure 7-18 and you will see three examples of sample and hold clocks vs data storage for multiplexed systems. The first two examples (shown in Figures 7-18A and 7-18B) are timing diagrams of the demo board. In these diagrams, the sample clock data is the data on path B of Figure 7-17. Likewise, the hold clock data is the data on path A.

In the Format Menu shown in Figure 7-16, path B data is the group associated with H2. Path B is always transparent, because hold clock H2 has been programmed don't care (i.e., pass always).

Path A, the data above path B, is transparent when its hold clock field is programmed to don't care. If path A is programmed to H1 and the current definition of H1 is LO=PASS (HI=HOLD), then the shaded area in Figure 7-18B represents the period when the presample latch is transparent and the white area represents the period when it is holding latched data.

Figure 7-19 shows the schematic for the demo board. As you can see, the counters are clocked on the rising edge of H1, and H1 is delayed relative to S1 by two gate delays. Also, remember that the counters are clocked into the state analyzer on the falling edge of S1.

Starting at the left side of the timing diagram in Figure 7-18B, you can see that shortly after the first rising edge of H1, the sample clock data (the lower 8 bits of the counter) changes to the value 57. The next falling edge of S1 (marked with the vertical line) clocks the sample and hold clock data into the state analyzer so that the displayed value is 5756. (This is the first value displayed in the column just past the far right-hand side of the timing diagram.) The 57 is the sample clock data and the 56 is the hold clock data. Why the hold clock data is 56 will become apparent as we go through the next cycle of the timing diagram.

Immediately after the first falling edge of S1, H1 goes low, causing the hold clock data presample latch to become transparent. Shortly after this latch has been enabled, the data in the latch becomes 57 as indicated. The 57 is the current value of the lower 8 bits of the counter.

Upon the next rising edge of H1, two things happen: the hold clock data is latched into the presample latch, and the counters are clocked. After a propagation delay from clock to counter is output, the counters increment to the value 58.

At this point the counters (and therefore the sample data path) have the value 58, while the hold clock presample latch has the value 57. The presample latch will hold the value 57 until H1 goes low. However, before this happens S1 goes low, clocking the value 5857 into the state analyzer. (The next four cycles of the timing diagram are left as an exercise for the reader.)

In Figure 7-18A, you can see that if the programmed polarity at H1 is reversed (in relation to Figure 7-18B), sample clock data and hold clock data will be the same at the time S1 clocks the data into the state analyzer. This is because both the sample and hold clock presample latches are transparent (enabled) before and during the S1 falling edge.
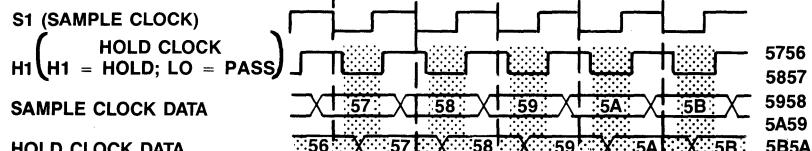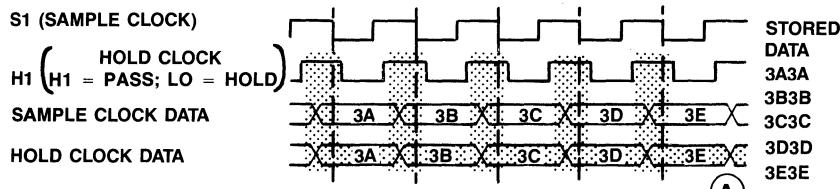
Now that you understand how programming the Format Menu in different ways can result in the state analyzer seeing different data, let's go on to the Trigger-Store Menu and learn how trigger specification separates one kind of data from another.
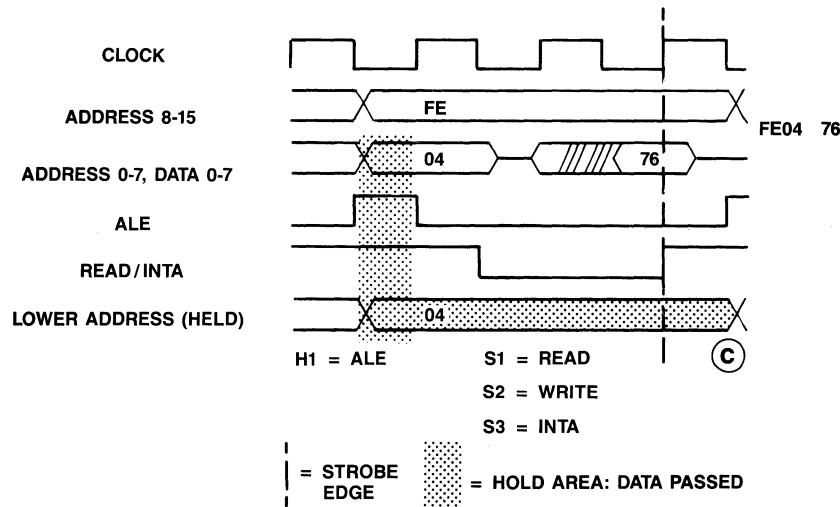
00842-072

TIMING EXAMPLES OF SAMPLE AND HOLD CLOCKS
VS DATA STORAGE FOR MULTIPLEXED SYSTEMS

S1 (SAMPLE CLOCK)

HOLD CLOCK
H1 (H1 = PASS; LO = HOLD)

SAMPLE CLOCK DATA

HOLD CLOCK DATA

STORED
DATA
3A3A
3B3B
3C3C
3D3D
3E3E

(A)

S1 (SAMPLE CLOCK)

HOLD CLOCK
H1 (H1 = HOLD; LO = PASS)

SAMPLE CLOCK DATA

HOLD CLOCK DATA

5756
5857
5958
5A59
5B5A

(B)

EXAMPLE: 8085 READ CYCLE

CLOCK

ADDRESS 8-15

ADDRESS 0-7, DATA 0-7

ALE

READ/INTA

LOWER ADDRESS (HELD)

FE04   76

(C)

H1 = ALE   S1 = READ
               S2 = WRITE
               S3 = INTA

| = STROBE   = HOLD AREA: DATA PASSED
   EDGE

■ Figure 7-18. Demonstration Timing Diagram.
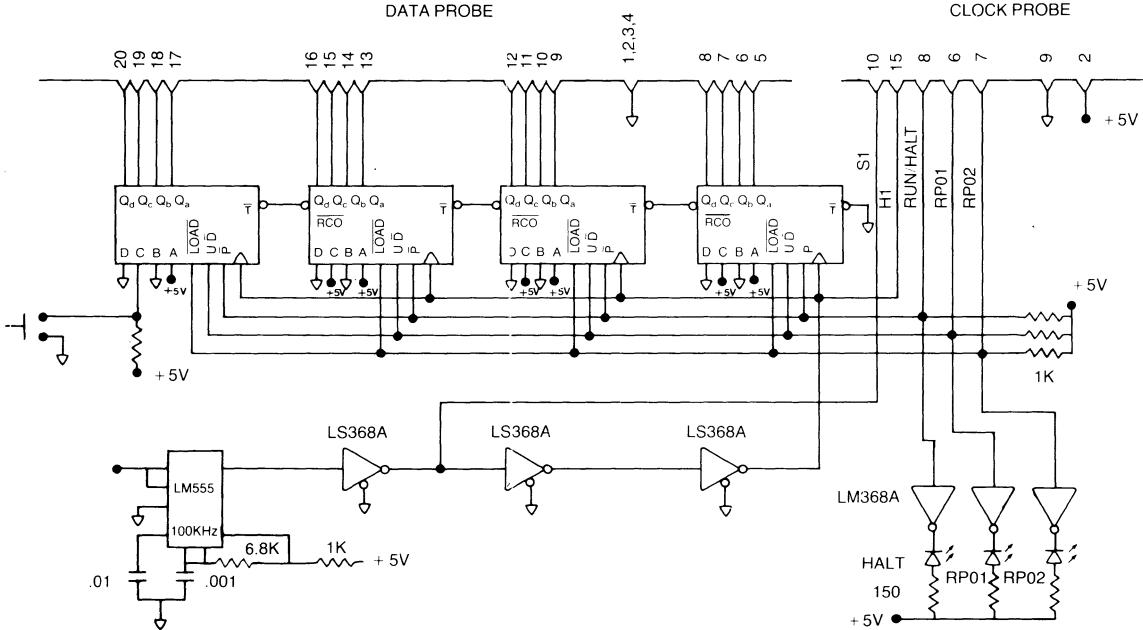
■ Figure 7-19. Schematic of the demo board.

████ THE TRIGGER-STORE MENU

In the Trigger-Store Menu the current trigger specification will detect differences in data patterns that result from changing how data is acquired in the Format Menu. (Refer back to the preceding sub-section for details.)

You don't need to read over the preceding subsection to follow the demo, but you do need to know the examples of Format Menu programming we used. To summarize, the clocking setup is:

1. MUX = No and hold clock = X.
2. MUX = Yes and hold clock = X.
3. MUX = Yes and hold clock = H1.

The data patterns associated with each mode are as follows:

1. MUX = No and hold clock = X. The data is a straight 16-bit counter that counts up or down, e.g.:
   5556
   5557
   5558
   5559

2. MUX = Yes and hold clock = X. The data is 8 bits of the counter, duplicated on both the upper and lower 8 bits of the displayed pattern, e.g.:
   3A3A
   3B3B
   3C3C
   3D3D

3. MUX = Yes and hold clock = H1. The data is as in (2), except that the count on the upper 8 bits leads the count on the lower 8 bits by one, e.g.:
   5756
   5857
   5958
   5A59

In addition to this information, you will need the Trigger-Trace Flowchart in Figure 7-10 and the information in Section 5 on how to use the Trigger-Store Menu.

Go to the Trigger-Store Menu by pressing M for Menus and T for Trigger-Store.

Your screen should look like Figure 7-20. For convenience, Figure 7-21 shows the entire specification.

```
TRIGGER-STORE MENU  Demo32  TRIG POS: DAY 0000 STOPPED at state Not =
STATE                                                           RPO CL
Start:  IF   anything  OCCURS 0001 TIMES THEN -------------> GOTO TstMx  3   1
                       NEUER  ------------------------> STORE

TstMx:  IF   Multiplex OCCURS BEFORE 0005 SAMPLE CLKS THEN -> GOTO Mux    3   1
        ELSEIF Non-Mux  OCCURS ANYTIME THEN ----------------> GOTO Nomux  0   1
               NEUER  ------------------------> STORE

Mux  :  IF   Shift     OCCURS BEFORE 1000 SAMPLE CLKS THEN -> GOTO Not =  2   1
        ELSE -------------------------------------------------> TRIGGER   3   1
             NEUER  ------------------------> STORE

Nomux:  IF  LowReset  OCCURS 0001 TIMES THEN -------------> GOTO Wait    0   0
        ELSEIF Hi-Reset OCCURS ANYTIME THEN ----------------> GOTO Cntup  0   0
               NEUER  ------------------------> STORE

- - - - - - - - - - - - - selected symbol - - - - - - - - - - - - - - - -
   NAME     Counter  EXT  CL
Acq 1234     1234     X    X


)Command: Display   Edit   Go   Menus   Scrollstates   enUir   ?=help  (SP)
```

■ **Figure 7-20. The Trigger-Store Menu.**

00842-075

```
TRIGGER-STORE MENU  Demo32  TRIG POS: ▓▓ ▓▓▓ STOPPED at state Not =
STATE                                                              RPO CL
Start:   IF   anything  OCCURS 0001 TIMES THEN --------------> GOTO TstMx  3   1

                        NEVER  ------------------------> STORE

TstMx:   IF   Multiplex OCCURS BEFORE 0005 SAMPLE CLKS THEN -> GOTO Mux    3   1
       ELSEIF Non-Mux   OCCURS ANYTIME THEN ----------------> GOTO Nomux   0   1
                        NEVER  ------------------------> STORE

Mux  :   IF   Shift     OCCURS BEFORE 1000 SAMPLE CLKS THEN -> GOTO Not =  2   1
       ELSE  ----------------------------------------------> TRIGGER      3   1
                        NEVER  ------------------------> STORE
Nomux:   IF   LowReset  OCCURS 0001 TIMES THEN --------------> GOTO Wait   0   1
       ELSEIF Hi-Reset  OCCURS ANYTIME THEN ----------------> GOTO Cntup   3   1
                        NEVER  ------------------------> STORE

Not =:   IF   anything  OCCURS 0001 TIMES THEN --------------> TRIGGER     2   1

                        NEVER  ------------------------> STORE

Wait :   IF   Hi-Reset  OCCURS 0001 TIMES THEN --------------> GOTO Cntup  3   1

                        NEVER  ------------------------> STORE

Cntup:   IF   Hi-Reset  OCCURS 0003 TIMES THEN --------------> GOTO DnCnt  2   1

                        NEVER  ------------------------> STORE

DnCnt:   IF   Hi-Reset  OCCURS 0002 TIMES THEN --------------> TRIGGER     3   1

                        NEVER  ------------------------> STORE
- - - - - - - - - - - - - - - selected symbol - - - - - - - - - - - - - - - -
   NAME    Counter  EXT  CL
Hi-Reset   5555     X    X


>Command: Display   Edit   Go   Menus   Scrollstates   enVir   ?=help  <SP>
```

■ Figure 7-21. The Trigger-Store Menu with all the states programmed.

Data values in the trigger-store specification are represented symbolically. Figure 7-22, a listing of the Symbol Menu, gives you a quick reference for the hexadecimal value of any word recognizer.
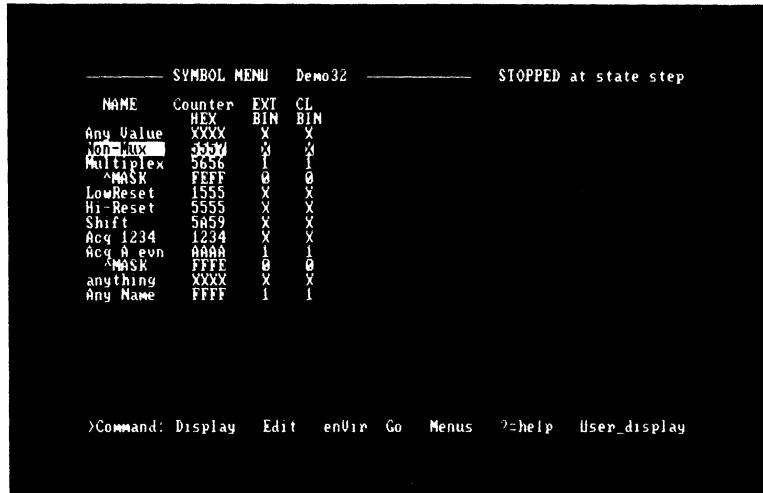
Perhaps the best way to look at the trigger-store specification is as a small high-level language program consisting of up to 15 procedures. Each procedure corresponds to the labels in the STATE column down the left-hand side of the specification (see Figure 7-21).

Each procedure (or STATE) consists of a number of lower level statements that allow you to jump to (GOTO) other procedures; to halt (TRIGGER); and to execute I/O as a function of the data monitored by the state analyzer's data probes (where Input = STORE and Output = Real-time Programmable Outputs).

At this point the analogy breaks down, because, in the analyzer's state machine, all the lower level instructions within each procedure can be executed in parallel (simultaneously), rather than serially as in a computer program.

7-30

On the Trigger-Trace Flowchart (Figure 7-10), the large boxes and diamonds correspond to the STATES in the trigger specification of Figure 7-21. The smaller boxes enclose the symbols used in the trigger specification. The data values next to the smaller boxes are the same as those indicated in the Symbol Menu in Figure 7-22.

00842-076

■ **Figure 7-22. The Symbol Menu.**

One last piece of information before we go through the trigger-store program. On the right-hand side of the screen there is a column labeled RPO. The values in this column determine the levels the two Real-time Programmable Outputs (RPOs) on the clock probe take upon leaving the state they are specified in. These RPOs are connected to the demo board in such a way that they control the up/down function and the load line of the counter. Table 7-1 lists the state the counter will take on for a given RPO programming.

■ **Table 7-1: RPO-Counter Programming**

RPO Value	Counter State
0	Load
1	Load
2	Count Down
3	Count

When you execute the Go command, the initial RPO value will be 0. When the counter is forced to load, the value it loads will be 5555 if the jumper is removed from the demo board, or 1555 if the jumper is installed.

We'll begin with state START in Figure 7-21. (Follow along on Figure 7-21 and your flowchart.) In this state, the first data value clocked into the state analyzer (anything=XXXX) will cause a jump to state TSTMX. It will also set the RPO value to 3, causing the counter to begin counting up from either 1555 or 5555 (depending on whether or not the jumper is installed).

In state TSTMX, the idea is to differentiate between data acquired in MUX mode and data acquired with MUX turned off. We know that in state START the counter was loaded with 1555 or 5555, and will now begin counting up. If (as explained previously) the For-mat Menu is in MUX mode, the upper and lower 8 bits of data will begin counting up simultaneously from either 5555 or 5655. One clock later their value will be 5656 or 5756.

These values correspond to the symbol Multiplex, and will cause the state machine to jump to the state MUX and continue to count up (RPO=3). If, however, the Format Menu has been used to turn MUX mode off, then the state analyzer will see a 16-bit counter begin to count up from either 1555 or 5555, eventually reaching the value of 5557 (Non-mux) before it counts through the Multiplex values. Non-mux will cause a jump to state NOMUX and will put the counter in a load condition (RPO=0).

We will follow the Multiplex path first and return later to the Non-Mux path.

State MUX will differentiate between data where the upper 8 bits equal the lower 8 bits, and data where the upper 8 bits have a value one greater than the lower 8. The state is looking for the Shift symbol (value 5A59) before 1000 counts have gone by.

We know that upon entering the state MUX, the count was either 5656 or 5756. Three clocks later it will be either 5A59 or 5959. If, on the Format Menu, the H1 hold clock is selected, the count value will be 5A59. The upper data bits and the lower data bits are not equal, and the state machine will jump to the state NOT=. If the hold clock is off (hold clock=X), then the value 5A59 will never occur, because the upper 8 bits will always equal the lower 8 bits. After 1000 clocks, the state analyzer will trigger from state MUX as directed.

State NOT= is being used as a marker. The state analyzer triggers as soon as it enters this state. Without looking at the acquired data, we know something about the sequence of data just by seeing the triggering end point. Marker states like this one are very useful in manufacturing test situations where the end points of tests can be states with labels such as Pass or Fail, depending on the sequence of data acquired.

Returning now to the Non-Mux path, state NOMUX determines whether or not the jumper is installed on the demo board. If the Low Reset symbol (value 1555) is detected, the jumper is installed and the state machine jumps to state WAIT, leaving the counter in load mode (RPO=0). If, however, the jumper is not installed, the Hi-Reset symbol (value 5555) will cause the state machine to jump to state CNTUP. Effectively, the state machine remains in state WAIT until the jumper is removed from the demo board.

In state CNTUP, the counter counts up until it rolls over twice and again reaches the count 5555 (Hi-Reset). Then the state machine jumps to state DNCNT and instructs the counter to count down (RPO=2).

In state DNCNT, the counter counts down until it rolls over once and again reaches the count 5555. At that time the state machine causes the analyzer to trigger.
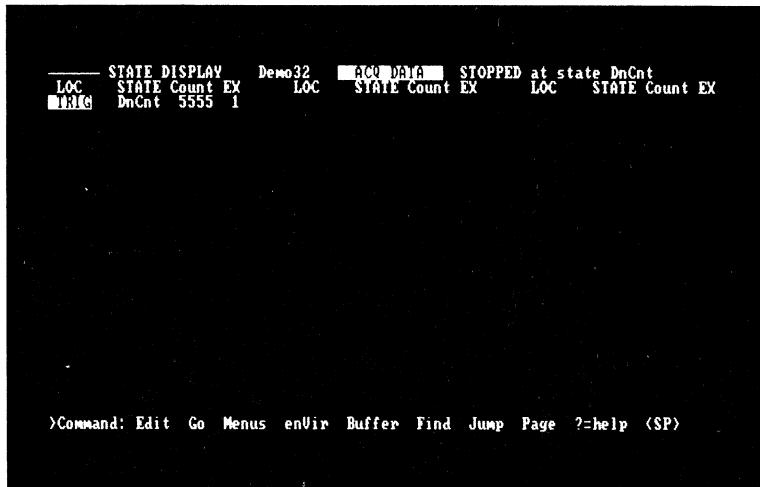
In each of the states in the trigger specification (shown in Figure 7-21), the state analyzer was directed to NEVER STORE. Even though the data was monitored and the state machine made decisions based upon this data (jumping, setting the RPOs, and triggering), no data was stored. No data, that is except the trigger word. Regardless of the store specification, the trigger word is always stored.
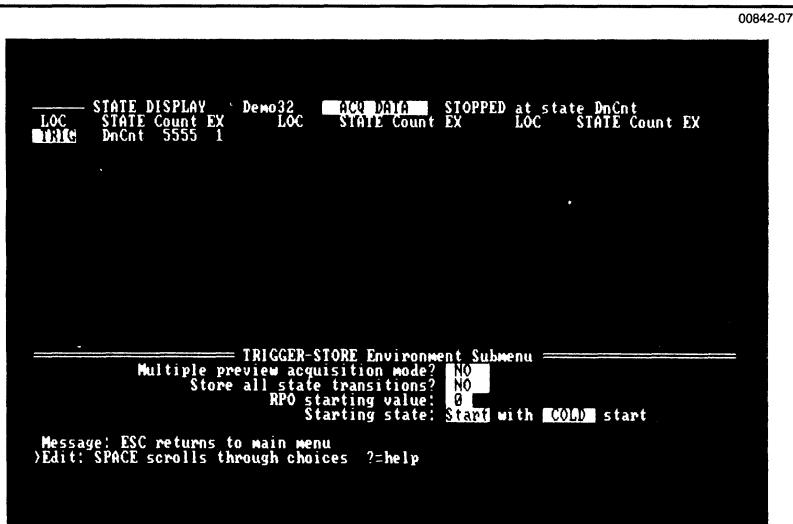
Let's try it. Execute the Go command. After the "STOPPED..."
message appears, enter the Display Menu. Your screen should now
look like Figure 7-23. The only value recorded was the trigger condi-
tion 5555 (Hi-Reset).

```
——————  STATE DISPLAY    Demo32   █ACQ DATA█  STOPPED at state DnCnt
  LOC     STATE  Count  EX      LOC     STATE Count EX      LOC     STATE Count EX
█TRIG█    DnCnt  5555   1



















>Command: Edit  Go  Menus  enVir  Buffer  Find  Jump  Page  ?=help  (SP)
```

■ **Figure 7-23. The State Display.**

Now, let's enter the Trigger-Store Environment Submenu. Press V for enVironment, then T for Trigger-Store. Your screen should now look like Figure 7-24.

■ Figure 7-24. The Trigger-Store Environment Submenu.

There are five programmable fields in the Trigger-Store Environment Submenu. The RPO Starting Value field specifies the encoded value that the two RPO lines assume when the Go command is executed. The Starting State field specifies the first state to be entered when the Go command is executed. (We will explain the field labeled COLD later when discussing single-step applications in conjunction with the run/stop line on the clock probe.) We will come back to the Multiple-Preview Acquisition Mode field in a moment. The Store All State Transitions field allows you to see all the states passed through on the way to triggering. Change this field now to YES, then press the escape key to re-enter the Display Menu.

Execute the Go command. Your screen should look like Figure 7-25. Note that the state sequence matches the flowchart.

00842-079

```
————— STATE DISPLAY    Demo32   ■ACQ DATA■  STOPPED at state DnCnt
  LOC    STATE Count EX      LOC    STATE Count EX      LOC    STATE Count EX
BEGIN   Start  5555  1
-0003   TstMx  5557  1
-0002   Nomux  5555  1
-0001   Cntup  5555  1
 TRIG   DnCnt  5555  1




 >Command: Edit  Go  Menus  enVir  Buffer  Find  Jump  Page  ?=help  <SP>
```

■ **Figure 7-25. All state transitions stored.**

Try it again with the jumper on. Install the jumper on the demo board, then execute the Go command. Remove the jumper when you see the "RUNNING at State WAIT" message. Your display should look like Figure 7-26.

00842-080
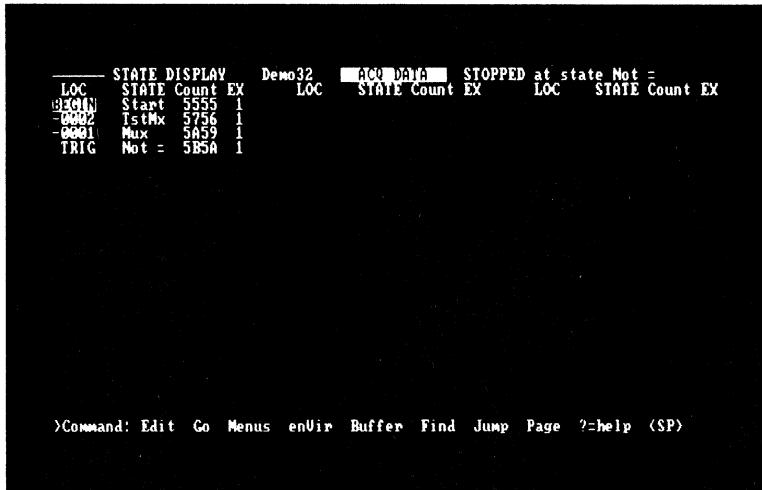
■ **Figure 7-26. With jumper on.**

Now, remove the jumper and try acquiring data after reprogramming the Format Menu to MUX mode with and without the H1 hold clock. After each trial return to the display and compare the state sequence with the flowchart. You should get displays like the ones shown in Figures 7-27 and 7-28.

00842-081


```
─────── STATE DISPLAY   Demo32  ▐ ACQ DATA ▌ STOPPED at state Not =
   LOC    STATE Count EX   LOC    STATE Count EX      LOC    STATE Count EX
 BEGIN    Start  5555  1
 -0002    IstMx  5756  1
 -0001    Mux    5A59  1
 TRIG     Not =  5B5A  1
```

```
 >Command: Edit  Go  Menus  enVir  Buffer  Find  Jump  Page  ?=help  <SP>
```

■ **Figure 7-27. MUX Mode with Hold Clock.**

00842-082

```
         STATE DISPLAY   Demo32    ACQ DATA    STOPPED at state Mux
   LOC     STATE Count EX    LOC     STATE Count EX      LOC     STATE Count EX
  BEGIN    Start 5555  1
 -0001     TstMx 5656  1
  TRIG     Mux   3F3F  1




  )Command: Edit  Go  Menus  enVir  Buffer  Find  Jump  Page  ?=help  <SP>
```

■ Figure 7-28. MUX Mode without Hold Clock.

Return now to the Trigger-Store Environment Submenu, and we'll learn about the Multiple-Preview Acquisition (MPA) mode.

MPA allows us to view the seven cycles immediately preceding every qualified acquisition. A qualified acquisition is defined as: any data stored as the result of the store specification within a state; the trigger word; or state transitions, if the Store All State Transitions field is set to YES.

Because the demonstration trigger specification now contains NEVER STORE in all states, the only qualified acquisitions will be the trigger word and state transitions. If we enable MPA mode, we will be able to see the seven unqualified cycles leading up to the trigger and state transitions.
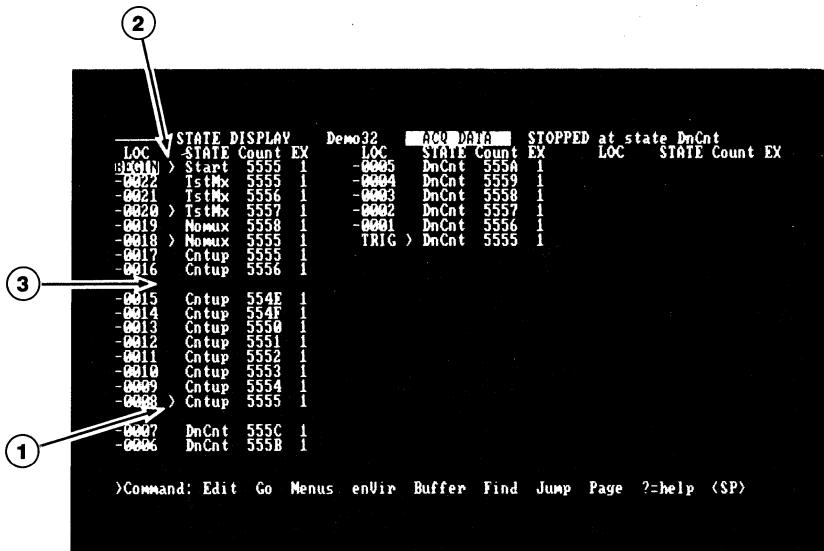
Enable MPA mode now by selecting YES in its field. Exit the Trigger-Store Environment Submenu by pressing escape, and re-enter the Display Menu.

Before executing the Go command, return the Format Menu to non-mux mode with the hold clock programmed to X. Execute the Go command and re-enter the Display Menu. Your screen should look like the one shown in Figure 7-29.

00842-083

■ Figure 7-29. The MPA Mode.

If you display does not look like Figure 7-29, but the trigger word is in state DNCNT, then enter the edit mode and move the cursor to the programmable field under the LOC column. Press B. Your screen should now look like Figure 7-29.

Note that all qualified acquisitions are marked with a right arrow (see callout 1 in Figure 7-29). The spaces between blocks of data indicate that an indeterminate number of cycles, which were not recorded, occurred between one block and another. (A cycle is the synchronous data presented to the state analyzer by the logical OR of the non-overlapping clocks, S1-S5, depending on which sample clocks you have connected.)

In general, a block of data will consist of one qualified acqui-
sition and its seven preceding cycles (e.g., the block associated with
the right arrow). However, if a qualified acquisition occurs before
seven unqualified cycles do, the qualified cycle will be marked
wherever it occurs within the blocks, and the block will be padded
out with cycles occurring immediately after the qualified cycle. This
will continue until the block is filled (with a total of eight acquisitions).

If more than one qualified acquisition occurs within a block,
all of them will be marked (see callout 2 in Figure 7-29).

In MPA, spaces between lines of data (see callout 3 in Figure
7-29) indicate that some number of unrecorded cycles occurred, and
lines of data without spaces represent contiguous cycles.

Now let's see how to use the trigger-position delay counter
both in and out of MPA mode.

Enter the Trigger-Store Environment Submenu and turn off
MPA mode. Then, enter the Trigger-Store Menu and press E for Edit
mode. Move the screen cursor to the Trigger Position (TRIG POS)
field at the top of the screen. (If MPA mode were on, the field would
not appear in reverse video, and you would not be able to change
it.) Once in this field, scroll through your choices. You should see DLY
0000, BEGIN, CENTER, and END as your options.

The trigger-position delay counter—the DLY field—is used to
specify how many acquisitions are to occur after the trigger. Thus,
DLY 0005 will cause five acquisitions to be stored after the trigger
line of the display. For convenience, BEGIN, CENTER, and END are
preprogrammed values. For a memory depth of 4096, these have
values of 4082, 2047, and 13 respectively.

Program the trigger position to DLY 0002. Then, use the
down cursor key to move state DNCNT onto the screen.

Now move the screen cursor into the NEVER field of state DNCNT. Scroll through your choices, stopping at IF ACQ 1234 or Acq A evn OCCUR THEN. At this point move the cursor into the Acq A evn field. Use the Control ($\wedge$ S) command to change the symbol selected in the field to Hi-Reset. The Trigger-Store Menu should now be programmed as indicated in Figure 7-30.

00842-084
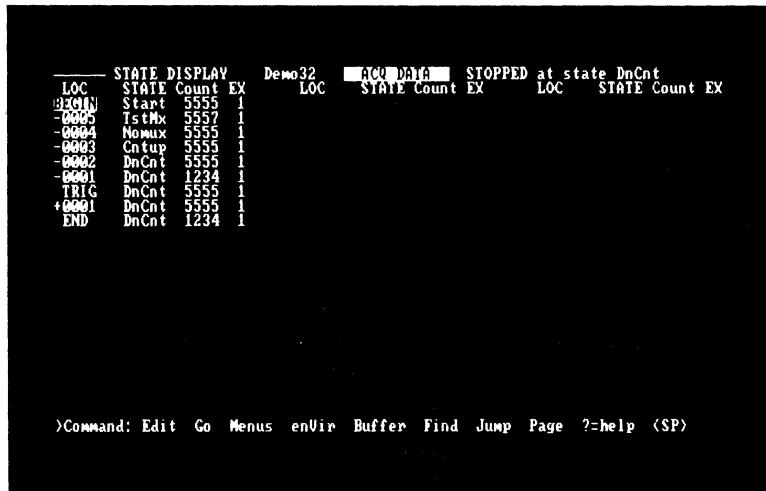
```
TRIGGER-STORE MENU  Demo32   TRIG POS: 0A4 0002 STOPPED at state DnCnt
STATE                                                              RPO CL
Not =:    IF    anything  OCCURS 0001 TIMES THEN -------------> TRIGGER   2   1
                          NEVER  ----------------------> STORE
Wait :    IF   Hi-Reset   OCCURS 0001 TIMES THEN -------------> GOTO Cntup  3   1
                          NEVER  ----------------------> STORE
Cntup:    IF   Hi-Reset   OCCURS 0003 TIMES THEN -------------> GOTO DnCnt  2   1
                          NEVER  ----------------------> STORE
DnCnt:    IF  Hi-Reset  OCCURS 0002 TIMES THEN -------------> TRIGGER     0   0
               IF  Acq 1234  OR   Hi-Reset  OCCUR THEN ---------> STORE
- - - - - - - - - - - - - - - selected symbol - - - - - - - - - - - - - -
   NAME    Counter  EXT  CL
Hi-Reset     5555    X    X

)Command: Display   Edit   Go   Menus   Scrollstates   enVir   ?=help  (SP)
```

■ **Figure 7-30. Programming the delay count.**

Execute the Go command and then enter the Display Menu. The data acquired should be as shown in Figure 7-31. Note that all state transitions were recorded and, in the state DNCNT, data values associated with the Acq 1234 and Hi-Reset symbols were acquired. Finally, note that there were exactly two acquisitions after the trigger line, as programmed in the trigger position delay count field.

00842-085


```
——————— STATE DISPLAY    Demo32   ▮ACQ DATA▮  STOPPED at state DnCnt
  LOC      STATE Count EX      LOC     STATE Count EX     LOC     STATE Count EX
 BEGIN    Start 5555  1
 -0005    IstMx 5557  1
 -0004    NoMux 5555  1
 -0003    Cntup 5555  1
 -0002    DnCnt 5555  1
 -0001    DnCnt 1234  1
  TRIG    DnCnt 5555  1
 +0001    DnCnt 5555  1
  END     DnCnt 1234  1



   >Command: Edit  Go  Menus  enVir  Buffer  Find  Jump  Page  ?=help  (SP)
```

■ **Figure 7-31. State Display—acquired data with delay count.**

██████ THE STATE DISPLAY

By this time you have used the Display Menu repeatedly, but we have not talked about many of the commands, nor about the Display Environment Submenu. If you have not already done so, read the *Display Menu* subsection in Section 5.

Now let's try using the Find command. Enter the Display Environment Submenu by pressing V for enVironment, then D for Display. Move the screen cursor to the Find Pattern field and scroll the Hi-Reset symbol into view. Now press the insert key. This will cause a Followed By field to appear. Again scroll the symbols through this field until the Hi-Reset symbol appears. Your screen should now look like Figure 7-32.

The find specification will look for a 5555, followed by a 5555 within 01 location every time the Find command is executed in the State Display.

00842-086

■ **Figure 7-32. The Find Command.**

Exit the Display Environment Submenu by pressing the escape key, then enter the Display Menu. In the current display (as shown in Figure 7-31), note that the first occurrence of the data sequence:
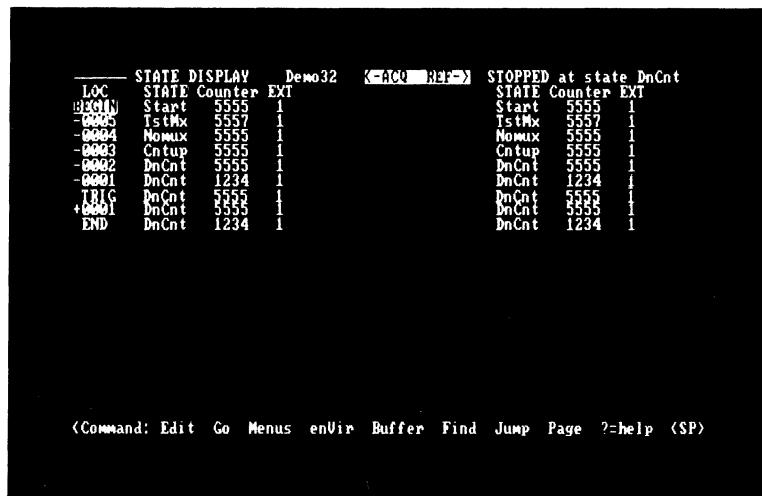
5555
5555

ends on line -0003. Execute the Find command. Note that line -0003 is now at the top of the screen. The next occurrence of the specified sequence ends on line +0001. Execute the Find command again. Now reverse the pattern search direction by pressing the < (less than) key. Try the Find command several times in this direction.

Now let's try using the compare acquisition to reference memory feature. First clean up the display by entering the LOC field and pressing B. Your display should look like the one in Figure 7-31. Select the < —ACQ REF— > option at the top of the display.

Escape from edit mode and execute the Ref< —acq command. Your screen should look like Figure 7-33.

00842-087

```
——————— STATE DISPLAY    Demo32  <-ACQ  REF->  STOPPED at state DnCnt
  LOC      STATE Counter EXT                    STATE Counter EXT
BEGIN     Start   5555   1                      Start   5555   1
-0005     TstMx   5557   1                      TstMx   5557   1
-0004     Nomux   5555   1                      Nomux   5555   1
-0003     Cntup   5555   1                      Cntup   5555   1
-0002     DnCnt   5555   1                      DnCnt   5555   1
-0001     DnCnt   1234   1                      DnCnt   1234   1
 TRIG     DnCnt   5555   1                      DnCnt   5555   1
+0001     DnCnt   5555   1                      DnCnt   5555   1
 END      DnCnt   1234   1                      DnCnt   1234   1




    <Command: Edit  Go  Menus  enVir  Buffer  Find  Jump  Page  ?=help  <SP>
```

■ **Figure 7-33. State Display—Ref< —acq.**

Next, re-enter the Display Environment Submenu. Move the cursor to the Compare Acq to Ref field and select YES.

7-45

To acquire data that differs slightly from the current reference data, run the state analyzer again, but this time with the jumper installed on the demo board. (Remember to remove the jumper when the "RUNNING in State WAIT" message appears.) After you execute the Go command, your display should look like the one in Figure 7-34.

00842-088

■ **Figure 7-34. Data acquired for comparison.**
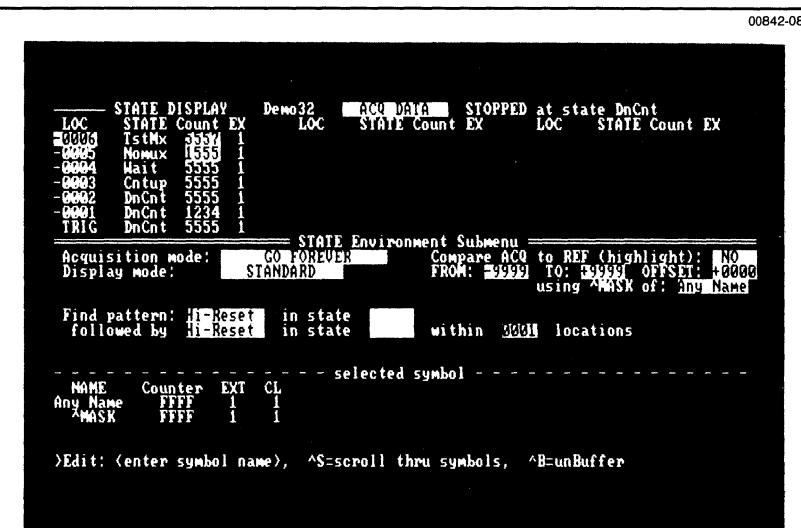
Notice two things about this display. First, the acquisition and reference displays are aligned in relation to their trigger words, unless you've specified an offset in the Display Environment Submenu. (The offset is currently 0.) Second, state labels are never compared. (Note line -0004 where the data is the same but the labels differ.)

Execute the Next_diff command. Note that the first difference (which is highlighted) moves to the top line. Try Next_diff a few more times. Also change direction and try it.

We will now go through an exercise that will teach you a few new things and allow you to practice much of what you have already learned. The objective is to step the counter a few counts at a time. This will illustrate how the state analyzer can be used to single- or multiple-step a microprocessor in your prototype.

First set up the Display Menu. Select the ACQ DATA option at the top of the display.

Now, enter the Display Environment Submenu and change the Acquisition Mode field to GO FOREVER. Change the Display Mode field to STANDARD, and turn off the Compare ACQ to REF field. Your Display Environment Submenu should now be programmed as shown in Figure 7-35.

00842-089


```
          STATE DISPLAY    Demo32    ACQ DATA    STOPPED at state DnCnt
    LOC    STATE Count EX    LOC    STATE Count EX    LOC    STATE Count EX
  -0006    TstMx  5557   1
  -0005    Nomux  5555   1
  -0004    Wait   5555   1
  -0003    Cntup  5555   1
  -0002    DnCnt  5555   1
  -0001    DnCnt  1234   1
   TRIG    DnCnt  5555   1
                                   STATE Environment Submenu
  Acquisition mode:      GO FOREVER        Compare ACQ to REF (highlight):  NO
  Display mode:          STANDARD          FROM: -9999  TO: +9999  OFFSET: +0000
                                                        using ^MASK of: Any Name

     Find pattern: Hi-Reset   in state           within  0001  locations
       followed by Hi-Reset   in state

   - - - - - - - - - - - - - - - selected symbol - - - - - - - - - - - - - - -
      NAME     Counter   EXT  CL
   Any Name     FFFF      1    1
     ^MASK      FFFF      1    1

  >Edit: <enter symbol name>,  ^S=scroll thru symbols,  ^B=unBuffer
```

■ **Figure 7-35. Setting up the display.**

Enter the Trigger-Store Menu. Set the trigger position to DLY 0000. Now move the cursor to the first State field. (This will probably be state START, but it depends on where you left the programming window the last time you were in this menu.)
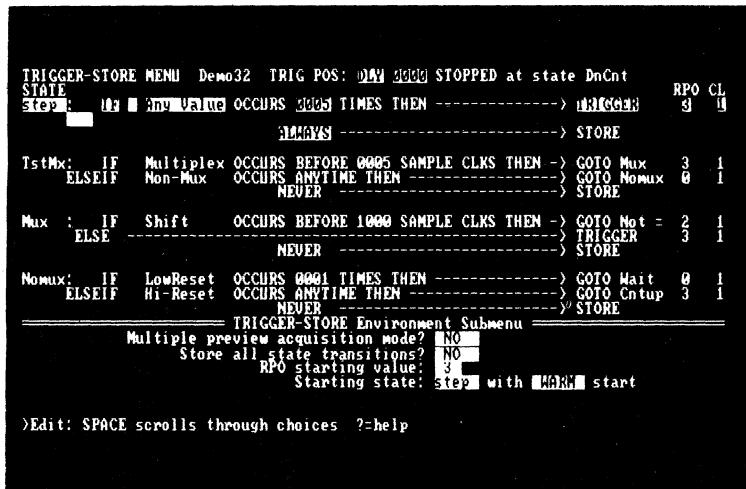
Press the insert key, and a new State field will appear. Name this state STEP. Move to the OCCURS field and enter 5.

Now escape from the edit mode and enter the Trigger-Store Environment Submenu. Turn off the MPA mode and the Store All State Transitions mode. Set the RPO Starting Value to 3, and set the Starting State to STEP with a WARM start. The Trigger-Store Environment Submenu should now be programmed as shown in Figure 7-36.

00842-090

■ Figure 7-36. Programming the Trigger-Store Menu.
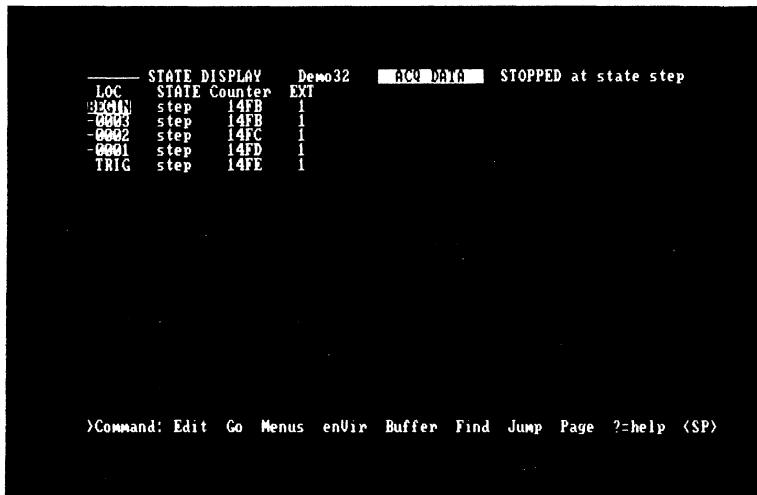
Because the starting RPO value is 3, these settings will cause the counter to begin counting up when you execute Go. The analyzer will start in state STEP and trigger after five clocks. The counter data associated with all five clocks will be stored, and because the delay counter is set to zero (DLY 0000), the counter will be disabled immediately when the trigger occurs. (Remember that the run/stop line on the clock probe is driving the counter's enable/disable input.)

WARM start will nearly always be the option to select when you want to use the state analyzer to stop a processor. The architecture of the analyzer is pipelined by one clock. This means that when data is displayed there is actually one more piece of data in the pipeline latch. The next time you command the analyzer to go, the first data into the machine is the old data left in the pipeline latch. With COLD start, internal logic causes this first piece of data to be discarded. With WARM start, the data is retained and clocked into the machine. Thus, when you're using the analyzer to single-step a processor, the data is one clock behind the processor.

Return to the Display Menu and execute the Go command. You should see a set of five data values repeated, as illustrated in Figure 7-37. Each element in the set of data is a count of five greater than the corresponding element in the previous set.

You may alternate between the Go and Stop commands to step the counter one block at a time. If you have not already done so, execute the Stop command.

00842-091

■ Figure 7-37. The State Display—counting by 5.

We're just about to move on to the Symbol Menu. First though, execute the Buffer command to save the data we will use in the next menu. Note the message at the bottom of the screen: "data from line -0004 saved in Symbol buffer."

Now go to the Symbol Menu.

████ THE SYMBOL MENU

Your screen should look like Figure 7-38.

The Symbol Menu is where data values are attached to symbols. There are two ways to do this. The first way is to create a symbol name and explicitly program each data column. The second way is to create a symbol name and then unbuffer the previously saved contents of the symbol buffer.

Let's try the second method now. Enter the edit mode. Move the cursor to any symbol name in the name column (on the far left). Now press the insert key. Type in any name you wish. While still in the name field, execute the unBuffer command, Control-B (^ B).

Now try creating a symbol entirely by hand. This is done in the same way as with the unBuffer method above, except that after you create the name, you will have to move into the data columns and enter the values by hand.

Move back to the name column. The commands available to you at the bottom of the screen include Toggle Mask. Try this command now. At this point you should read the explanation of the mask in the *Symbol Menu* subsection of Section 5.

The mask technique will allow you to specify bits within a digit (i.e., a hexadecimal or octal digit) as don't care, without having to expand the group to binary.
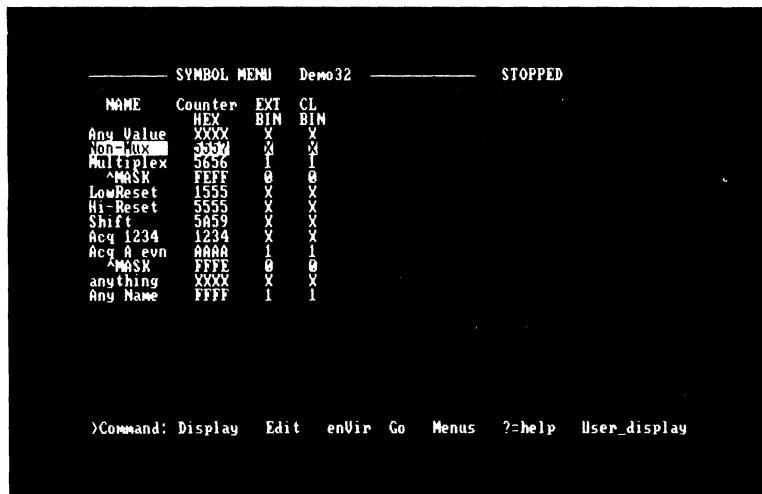
As a final exercise (especially if you still find the concept of the mask difficult) go to the Format Menu and change the radix of the Counter group to binary. Now return to the Symbol Menu and enter the edit mode.

Move the screen cursor to the Multiplex symbol's Name field. If a mask is not displayed below Multiplex, execute the Toggle Mask command to bring it into view. Note that there is only one zero in the mask. Now toggle the mask on and off slowly while you watch the relationship between the zero in the mask and the don't care bit in the Multiplex symbol. Note that only one don't care bit appears.

Now turn off the mask and translate the binary value of the Multiplex symbol into hexadecimal by hand. You should get 5656 or 5756.

Return to the Format Menu and change the Counter group's radix back to HEX. Return to the Symbol Menu. Note that the value displayed for the Multiplex symbol is 5?56. Enter edit mode and toggle the mask. Note that the question mark becomes a 6 and the mask digit below this 6 is E. The position at the zero bit value within the E marks the position of the don't care bit in the 6 above it.

00842-092

```
———————— SYMBOL MENU   Demo32  ———————        STOPPED

     NAME      Counter  EXT  CL
                HEX     BIN  BIN
  Any Value     XXXX     X    X
  Non-Mux       5556     X    X
  Multiplex     5656     1    1
   ^MASK        FEFF     0    0
  LowReset      1555     X    X
  Hi-Reset      5555     X    X
  Shift         5A59     X    X
  Acq 1234      1234     X    X
  Acq A evn     AAAA     1    1
   ^MASK        FFFE     0    0
  anything      XXXX     X    X
  Any Name      FFFF     1    1



   >Command: Display   Edit   enVir  Go   Menus   ?=help   User_display
```

■ **Figure 7-38. The Symbol Menu using a mask.**

ASCII CHARACTERS

0	NUL	32	(blank space)	64	@	96	'
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	ˇ	124	|
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	—	127	DEL

■■■■ DATA FILE FORMATS

This appendix explains the data structures and formats used in the I/O Menu's DATA files. These files are created whenever you save the contents of reference memory on disk.

Pascal Declaration of State Acquisition/Reference Data Type

Acq_Word = RECORD CASE INTEGER OF {type of 80 bit word of acq data}
      1: (int : ARRAY [0..4] of INTEGER); {5 words of data}
      2: (bite : ARRAY [0..9] of BYTE); {in byte form}

Acq_Data = RECORD {type of 96 bit-word read from acq hardware}
      misc   : WORD; {1 word of flag bits}
      data   : Acq-Word; {5 words of acq data (80 channels)}
      time_tag : INTEGER 4; {reserved for Time Stamp data}

Data File Formats

Bytes 0 to 511 are reserved for use by Northwest Instrument Systems, Inc.

All bytes between 512 and the end of the file, inclusive, are available to the user. The contents of these bytes are shown below.

■ **Table B-1: Byte Definitions**

Public Name	Byte	Contents
ref_beg	512 513	Low byte of beginning reference location. High byte of beginning reference location. Integer number from −4096 to +4096. This number tells where the start of state data is with respect to the trigger location (0).
ref_end	514 515	Low byte of ending reference location. High byte of ending reference location. Integer number from −1 to +8192. This number tells where the end of state data is with respect to the trigger location (0).
ref_mpa	516	MPA mode flag. Value for MPA mode: 1 = on, 0 = off.
saved_pod_count	517 518	Low byte of saved_pod_count. High byte of saved_pod_count. An integer from 1 to 5, which records the number of 16 channel pods available in the state analyzer system.
first_mux	519 520	Low byte of first_mux. High byte of first_mux. First_mux is an integer from 0 to 10. It indicates which byte of the data marks the boundary between non-mux and mux data bytes.
ref_has_time	521	A Boolean variable that indicates whether or not the saved reference data contains time data from the Time Stamp (Option 003). 0 = no time data present, 1 = time data present.
ref_ts_flgs	522	T_overflow. A Boolean variable that indicates whether or not the Time Stamp counter overflowed while reference data was acquired. 0 = no overflow, 1 = overflow.
	523	T_base_mode. Time Stamp timebase clock source used while reference data was acquired. 0 = internal 20 MHz clock, 1 = external clock, 2 = ISA sample clock.
	524	Ext_period. Time Stamp external timebase clock period. An integer from 2 to 14 indicates the external clock period used while reference data was acquired. 2 = 50 ns, 3 = 100 ns, 4 = 200 ns, 5 = 500 ns...14 = 500 μns.
	525	Reserved for future use by Northwest Instrument Systems, Inc.

Continued

■ Table B-1 (cont): Byte Definitions

Public Name	Byte	Contents
sla_ref_mode	526	An integer which records the time-alignment format that is used for aligning the state data with the data from the Model 2200 Interactive Timing Analyzer.
r_align_loc	527 528	Low byte of the reference alignment location. High byte of the reference alignment location: An integer which records the reference alignment location for aligning the state data with the data from the Model 2200 Interactive Timing Analyzer.
st_ref_name[0]	529	State 1 name header. This is the first of 15 groups of 6 bytes each. The first byte in each group is the header and tells how long the string will be. The next 5 bytes represent the letters that compose the state in ASCII.
	530 531 532 533 534	First letter of state name. Second letter of state name. Third letter of state name. Fourth letter of state name. Fifth letter of state name.
st_ref_name[14]	613 614 615 616 617 618	State 15 name header. First letter of state name. Second letter of state name. Third letter of state name. Fourth letter of state name. Fifth letter of state name.

Physical Data Structure

Starting with byte 619, the words of reference data are stored in the format shown below. The first word of this format corresponds to the data stored at location ref_beg.

Flag Byte
Data Byte 0 ⎫
Data Byte 1 ⎪ The number of data bytes is
Data Byte 2 ⎬ determined by the following formula:
Data Byte 3 ⎪
Data Byte 4 ⎨ saved_pod_count * 2
Data Byte 5 ⎪
Data Byte 6 ⎫ where saved_pod_count equals
Data Byte 7 ⎬ the number of pods that were avail-
Data Byte 8 ⎭ able when reference data was saved.
Data Byte 9
Time Byte 0 ⎫
Time Byte 1 ⎬ The time bytes are reserved for time
Time Byte 2 ⎨ data acquired via the Time Stamp.
Time Byte 3

As shown above, each word of the stored reference data is comprised of the flag byte and the (saved_pod_count * 2) data bytes. Each word may also contain four time bytes, depending on whether or not the Time Stamp (Option 003) was used to acquire time data. The ref_has_time flag indicates whether or not the time bytes are present.

Physical Data Structure for Non-Multiplexed Probes. The following table shows the bit structure used for reference data that has been acquired with non-multiplexed probes.

■ **Table B-2: Bit Structure for Non-Multiplexed Probes**

Byte Number	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Flag Byte	Qual Acq	Ring Full	Cross Link	Ext Input	State Bit 3	State Bit 2	State Bits 1	State Bits 0
Probe 1:								
Data Byte 0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Data Byte 1	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
Probe 2:								
Data Byte 2	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
Data Byte 3	CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24

Continued

■ Table B-2 (cont): Bit Structure for Non-Multiplexed Probes

Byte Number	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Probe 3:								
Data Byte 4	CH39	CH38	CH37	CH36	CH35	CH34	CH33	CH32
Data Byte 5	CH47	CH46	CH45	CH44	CH43	CH42	CH41	CH40
Probe 4:								
Data Byte 6	CH55	CH54	CH53	CH52	CH51	CH50	CH49	CH48
Data Byte 7	CH63	CH62	CH61	CH60	CH59	CH58	CH57	CH56
Probe 5:								
Data Byte 8	CH71	CH70	CH69	CH68	CH67	CH66	CH65	CH64
Data Byte 9	CH79	CH78	CH77	CH76	CH75	CH74	CH73	CH72

Physical Data Structures for Multiplexed Probes. For multiplexed probes, data is organized as a list of bytes from the non-multiplexed pods, followed by a list of bytes from multiplexed pods. The first_mux variable defines the first byte of the multiplexed part.

The following example shows the relationship between multiplexed probes in the Format Menu and the position of the data bytes within the reference data structures:

```
uSLOT  MUX  HOLD CLK   5432109876543210
  3    NO        X      [ byte 1 | byte 0 ]    non-mux
  4    YES       X              [ byte 2 ]     non-mux
                               [ byte 6 ]      mux        <— first_mux
  5    YES       X              [ byte 3 ]     non-mux
                               [ byte 7 ]      mux
  6    NO        X      [ byte 5 | byte 4 ]    non-mux
```

■■■■ CLOCKING EXAMPLES

This appendix provides several examples of how to use the state analyzer's clocking to acquire data from a microprocessor under test. Each example shows how to connect the data and clock channels to the microprocessor pins, and how to set up the Format Menu.

The microprocessors used for the examples include:

Z80
8085
8086
8088
68000

If you are attempting to acquire data from a microprocessor that is not discussed here, use the following examples as a guide.
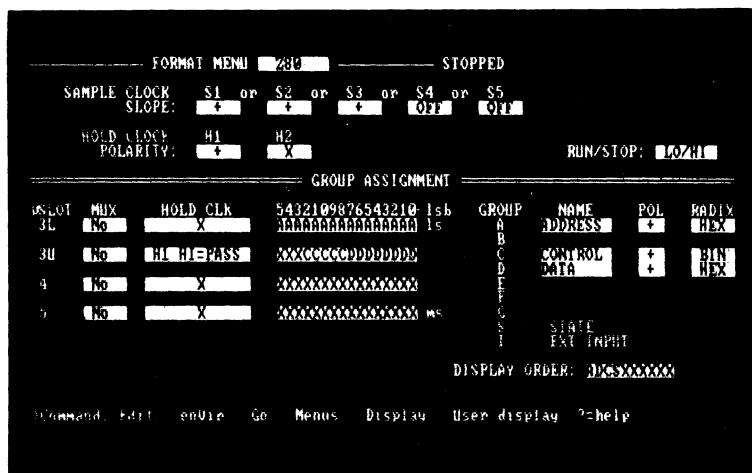
All of the above microprocessors are also supported by the state analyzer's Mnemonic Disassembler kits. These kits provide a single-plug probe connector, as well as preprogrammed disassembler software. For details, see *Related Products and Options* at the front of this manual.

███████ Z80 CLOCKING CONNECTIONS
(using 32 channels)

μSlot	ISA LEAD	Z80	PIN#
3	D0	A0	30
	D1	A1	31
	D2	A2	32
	D3	A3	33
	D4	A4	34
	D5	A5	35
	D6	A6	36
	D7	A7	37
	D8	A8	38
	D9	A9	39
	D10	A10	40
	D11	A11	1
	D12	A12	2
	D13	A13	3
	D14	A14	4
	D15	A15	5
4	D0	D0	14
	D1	D1	15
	D2	D2	12
	D3	D3	8
	D4	D4	7
	D5	D5	9
	D6	D6	10
	D7	$\overline{D7}$	13
	D8	$\overline{BUSACK}$	23
	D9	$\overline{WR}$	22
	D10	$\overline{IORQ}$	20
	D11	$\overline{M1}$	27
CLOCK	S1	$\overline{RD}$	21
	S2	$\overline{WR}$	22
	S3	$\overline{IORQ}$	20
	S4		
	S5		
	H1	CLK	6
	H2	LEAVE UNCONNECTED	

Z80 FORMAT MENU SETUP
(using 32 channels)

00842-093
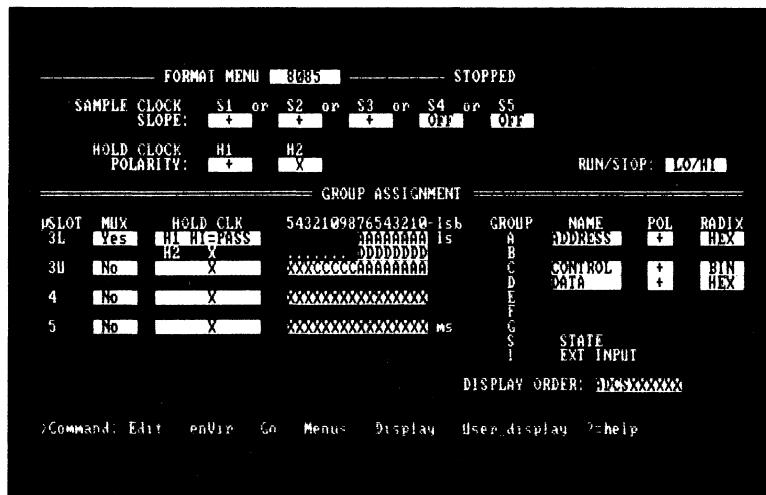
████ 8085 CLOCKING CONNECTIONS
(using 32 channels)

μSlot	ISA LEAD	8085	PIN#
1	D0	AD0	12
	D1	AD1	13
	D2	AD2	14
	D3	AD3	15
	D4	AD4	16
	D5	AD5	17
	D6	AD6	18
	D7	AD7	19
2	D0	A8	21
	D1	A9	22
	D2	A10	23
	D3	A11	24
	D4	A12	25
	D5	A13	26
	D6	A14	27
	D7	A15	28
	D8	S0	29
	D9	S1	33
	D10	IO/$\overline{M}$	34
	D11	$\overline{WR}$	31
	D12	HLDA	38
CLOCK	S1	$\overline{RD}$	32
	S2	$\overline{WR}$	31
	S3	$\overline{INTA}$	11
	S4		
	S5		
	H1	ALE	30
	H2	LEAVE UNCONNECTED	

8085 FORMAT MENU SETUP
(using 32 channels)
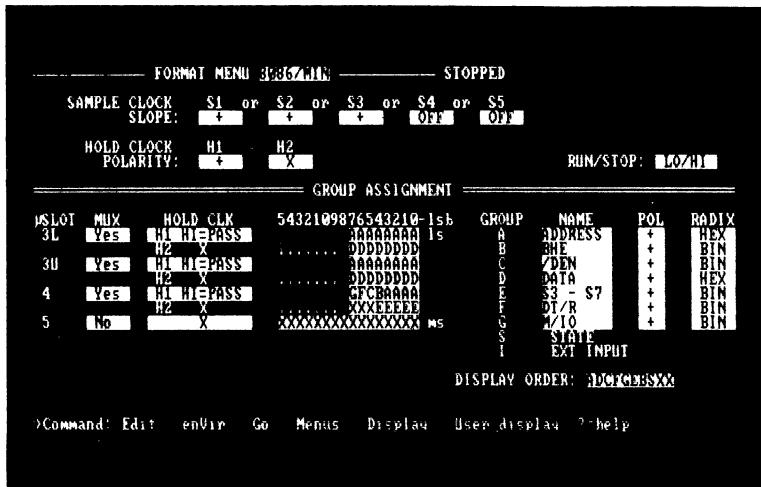
8086, MIN MODE, CLOCKING CONNECTIONS (using 48 channels)

μSlot	ISA LEAD	8086	PIN#
1	D0	AD0	16
	D1	AD1	15
	D2	AD2	14
	D3	AD3	13
	D4	AD4	12
	D5	AD5	11
	D6	AD6	10
	D7	AD7	9
2	D0	AD8	8
	D1	AD9	7
	D2	AD10	6
	D3	AD11	5
	D4	AD12	4
	D5	AD13	3
	D6	AD14	2
	D7	AD15	39
3	D0	A16/S3	38
	D1	A17/S4	37
	D2	A18/S5	36
	D3	A19/S6	35
	D4	$\overline{BHE}$/S7	34
	D5	$\overline{DEN}$	26
	D6	DT/$\overline{R}$	27
	D7	M/$\overline{IO}$	28
CLOCK	S1	$\overline{RD}$	32
	S2	$\overline{WR}$	29
	S3	$\overline{INTA}$	24
	S4		
	S5		
	H1	ALE	25
	H2	LEAVE UNCONNECTED	

8086, MIN MODE, FORMAT MENU SETUP (using 48 channels)

00842-095

```
------------- FORMAT MENU 8086/MIN ------------- STOPPED

SAMPLE CLOCK    S1  or  S2  or  S3  or  S4  or  S5
       SLOPE:   [+]    [+]    [+]   [OFF]  [OFF]

HOLD CLOCK      H1      H2
   POLARITY:   [+]    [X]                        RUN/STOP: [LO/HI]

================== GROUP ASSIGNMENT ==================
µSLOT  MUX    HOLD CLK    5432109876543210-lsb   GROUP   NAME      POL   RADIX
 3L    [Yes]  [H1 HI-PASS]  AAAAAAAA ls            A     ADDRESS    +    HEX
              H2   X        DDDDDDDD                B     BHE        +    BIN
 3U    [Yes]  [H1 HI-PASS]  AAAAAAAA               C     /DEN       +    BIN
              H2   X        DDDDDDDD                D     DATA       +    HEX
 4     [Yes]  [H1 HI-PASS]  GFCBAAAA               E     S3 - S7    +    BIN
              H2   X        XXXEEEED               F     DT/R       +    BIN
 5     [No]        X       XXXXXXXXXXXXXXXXX ms    G     M/IO       +    BIN
                                                   S     STATE
                                                   I     EXT INPUT

                                    DISPLAY ORDER: [ADCFGEBSXX]

>Command: Edit  enVir  Go  Menus  Display  User display  ?-help
```

8086, MAX MODE, CLOCKING CONNECTIONS
(using 48 channels)

μSlot	ISA LEAD	8086	PIN#	8288	PIN#
1	D0	AD0	16		
	D1	AD1	15		
	D2	AD2	14		
	D3	AD3	13		
	D4	AD4	12		
	D5	AD5	11		
	D6	AD6	10		
	D7	AD7	9		
2	D0	AD8	8		
	D1	AD9	7		
	D2	AD10	6		
	D3	AD11	5		
	D4	AD12	4		
	D5	AD13	3		
	D6	AD14	2		
	D7	AD15	39		
3	D0	A16/S3	38		
	D1	A17/S4	37		
	D2	A18/S5	36		
	D3	A19/S6	35		
	D4	$\overline{BHE}$/S7	34		
	D5	$\overline{S0}$	26		
	D6	$\overline{S1}$	27		
	D7	$\overline{S2}$	28		
CLOCK	S1			$\overline{MRDC}$	7
	S2			$\overline{MWTC}$	9
	S3			$\overline{IORC}$	13
	S4			$\overline{IOWC}$	11
	S5			$\overline{INTA}$	14
	H1			ALE	5
	H2		LEAVE UNCONNECTED		

8086, MAX MODE, FORMAT MENU SETUP
(using 48 channels)

00842-096

```
―――――――― FORMAT MENU 8086/MAX ―――――― STOPPED

   SAMPLE CLOCK    S1   or   S2   or   S3   or   S4   or   S5
         SLOPE:    ↑        ↑        ↑        ↑        ↑

   HOLD CLOCK    H1       H2
       POLARITY:  ↑        X                      RUN/STOP: LO/HI

―――――――――――――――――― GROUP ASSIGNMENT ――――――――――――――――――

µSLOT  MUX     HOLD CLK    5432109876543210-lsb  GROUP   NAME      POL  RADIX
  3L   Yes     H1 HI-PASS  AAAAAAAA ls             A     ADDRESS    ↑   HEX
               H2    X     DDDDDDDD                B     BHE        ↑   BIN
  3U   Yes     H1 HI-PASS  AAAAAAAA                C     S0-S1-S2   ↑   BIN
               H2    X     DDDDDDDD                D     DATA       ↑   HEX
  4    Yes     H1 HI-PASS  CCCBAAAA                E     S3 - S7    ↑   BIN
               H2    X     XXXEEEEE                F
  5    No         X        XXXXXXXXXXXXXXXXX ms    G
                                                  S     STATE
                                                  I     EXT INPUT

                                    DISPLAY ORDER: ADCEBSXXXX

>Command: Edit   enVir   Go   Menus   Display   User display   ?=help
```
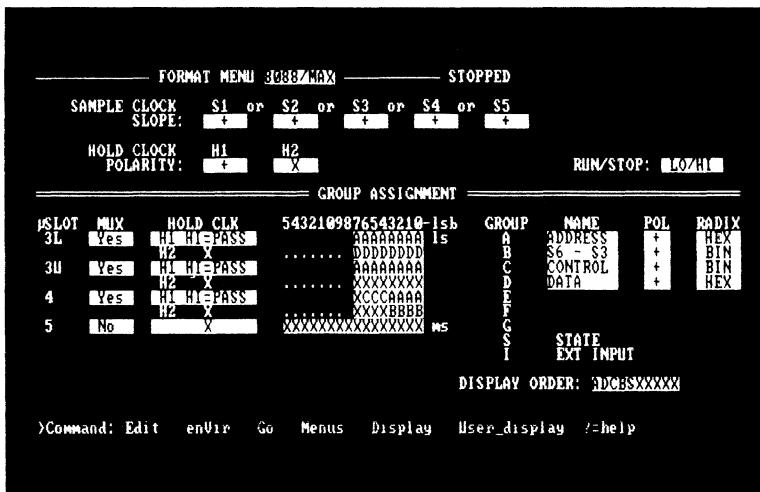
■■■■ 8088, MAX MODE, CLOCKING CONNECTIONS
(using 48 channels)

μSLOT	ISA LEAD	8088	PIN#	8288	PIN#
3	D0	AD0	16		
	D1	AD1	15		
	D2	AD2	14		
	D3	AD3	13		
	D4	AD4	12		
	D5	AD5	11		
	D6	AD6	10		
	D7	AD7	9		
4	D0	A8	8		
	D1	A9	7		
	D2	A10	6		
	D3	A11	5		
	D4	A12	4		
	D5	A13	3		
	D6	A14	2		
	D7	A15	39		
	D8	A16	38		
	D9	A17	37		
	D10	A18	36		
	D11	A19	35		
	D12	$\overline{S0}$	26		
	D13	$\overline{S1}$	27		
	D14	$\overline{S2}$	28		
CLOCK	S1			$\overline{MRDC}$	7
	S2			$\overline{MWTC}$	9
	S3			$\overline{IORC}$	13
	S4			$\overline{IOWC}$	11
	S5			$\overline{INTA}$	14
	H1			ALE	5
	H2			LEAVE UNCONNECTED	

8088, MAX MODE, FORMAT MENU SETUP
(using 48 channels)

00842-097

```
─────────────── FORMAT MENU 8088/MAX ─────────── STOPPED

   SAMPLE CLOCK    S1   or   S2   or   S3   or   S4   or   S5
         SLOPE:    [ + ]    [ + ]    [ + ]    [ + ]    [ + ]

   HOLD CLOCK     H1       H2
      POLARITY:   [ + ]    [ X ]                    RUN/STOP:  LO/HI

═══════════════════════ GROUP ASSIGNMENT ═══════════════════

µSLOT  MUX     HOLD CLK    543210987654321Ø-lsb  GROUP   NAME      POL  RADIX
  3L   Yes     H1 HI=PASS        AAAAAAAA ls       A     ADDRESS    +    HEX
               H2    X    ....... DDDDDDDD          B     S6 - S3    +    BIN
  3U   Yes     H1 HI=PASS        AAAAAAAA           C     CONTROL    +    BIN
               H2    X    ....... XXXXXXXX          D     DATA       +    HEX
  4    Yes     H1 HI=PASS        XCCCAAAA           E
               H2    X          XXXXBBBB            F
  5    No          X     XXXXXXXXXXXXXXXXX ms       G     STATE
                                                    S     EXT INPUT
                                                    I
                                      DISPLAY ORDER: ADCBSXXXXX

)Command: Edit    enVir   Go   Menus   Display   User_display   /=help
```

8088, MAX MODE CLOCKING
CONNECTIONS
(using 32 channels)

μSlot	ISA LEAD	8088	PIN#	8288	PIN#
1	D0	AD0	16		
	D1	AD1	15		
	D2	AD2	14		
	D3	AD3	13		
	D4	AD4	12		
	D5	AD5	11		
	D6	AD6	10		
	D7	AD7	9		
2	D0	A8	8		
	D1	A9	7		
	D2	A10	6		
	D3	A11	5		
	D4	A12	4		
	D5	A13	3		
	D6	A14	2		
	D7	A15	39		
3	D0	A16/S3	38		
	D1	A17/S4	37		
	D2	A18/S5	36		
	D3	A19/S6	35		
	D4	$\overline{S0}$	26		
	D5	$\overline{S1}$	27		
	D6	$\overline{S2}$	28		
CLOCK	S1			$\overline{MRDC}$	7
	S2			$\overline{MWTC}$	9
	S3			$\overline{IORC}$	13
	S4			$\overline{IOWC}$	11
	S5			$\overline{INTA}$	14
	H1			ALE	5
	H2	LEAVE UNCONNECTED			

8088, MAX MODE, FORMAT MENU SETUP
(using 32 channels)

00842-098

██████ 68000 CLOCKING CONNECTIONS (using 48 channels)
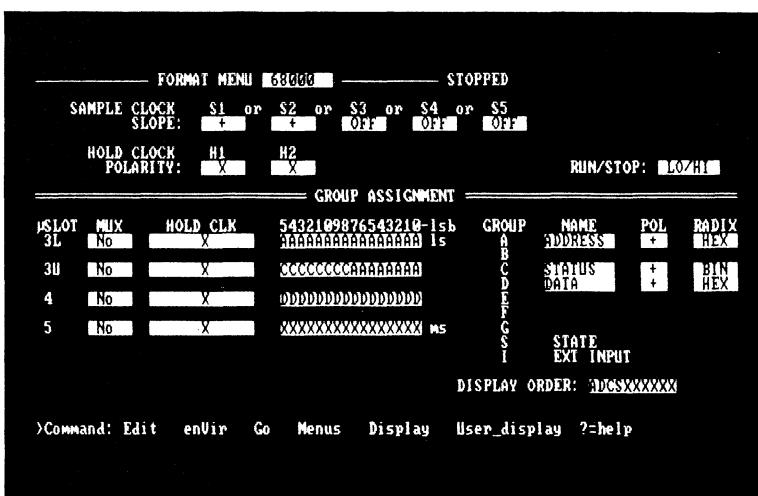
µSLOT	ISA LEAD	68000	PIN#
3	D0	GND	53
	D1	A1	29
	D2	A2	30
	D3	A3	31
	D4	A4	32
	D5	A5	33
	D6	A6	34
	D7	A7	35
	D8	A8	36
	D9	A9	37
	D10	A10	38
	D11	A11	39
	D12	A12	40
	D13	A13	41
	D14	A14	42
	D15	A15	43
4	D0	A16	44
	D1	A17	45
	D2	A18	46
	D3	A19	47
	D4	A20	48
	D5	A21	50
	D6	A22	51
	D7	A23	52
	D8	$\overline{\text{BERR}}$	22
	D9	$\overline{\text{BGACK}}$	12
	D10	$\overline{\text{UDS}}$	7
	D11	$\overline{\text{LDS}}$	8
	D12	$\overline{\text{R/W}}$	9
	D13	FC0	28
	D14	FC1	27
	D15	FC2	26
5	D0	D0	5
	D1	D1	4
	D2	D2	3
	D3	D3	2
	D4	D4	1
	D5	D5	64
	D6	D6	63
	D7	D7	62
	D8	D8	61

■■■■ 68000 CLOCKING CONNECTIONS (cont.)

µSLOT	ISA LEAD	68000	PIN #
5 (cont)	D9	D9	60
	D10	D10	59
	D11	D11	58
	D12	D12	57
	D13	D13	56
	D14	D14	55
	D15	D15	54
CLOCK	S1	LDS	8
	S2	$\overline{UDS}$	7
	S3	NOT CONNECTED	
	S4	NOT CONNECTED	
	S5	NOT CONNECTED	
	H1	NOT CONNECTED	
	H2	NOT CONNECTED	

■■■■ 68000 FORMAT MENU SETUP (using 48 channels)

00842-099

■■■ INDEX

C

D (continued)

H (continued)

I

O

P

S (continued)