T.O. 33-D7-32-12-8-1

DISTRIBUTION STATEMENT — Distribution authorized to U.S. Government agencies only for administrative or operational use (effective date is date of this manual). Other requests for this document must be referred to San Antonio ALC/MMEDT, Kelly AFB, TX 78241-5000.

THIS MATERIAL MAY BE REPRODUCED BY OR FOR THE U.S. GOVERNMENT PURSUANT TO THE COPYRIGHT LICENSE UNDER THE (DFAR) CLAUSE AT 52.227-7013 (15 MAY 1987).

HANDLING AND DESTRUCTION NOTICE — Comply with distribution statement and destroy by any method that will prevent disclosure of contents or reconstruction of the document.
Form No. A260 7/89

9100A Series

Technical User's Manual

P/N 813832
APRIL 1987 Rev. 3, 12/88

FLUKE

LIMITED WARRANTY

John Fluke Mfg. Co., Inc. (Fluke) warrants your 9100/9105A to be free from defects in material and workmanship under normal use and service for 90 days from the date of shipment. Software and firmware products are provided "AS IS." We do not warrant that software or firmware products will be error free, operated without interruption or that all errors will be corrected. This warranty extends to you if you are the original purchaser and does not apply to fuses, batteries or any product which, in our sole opinion, has been subject to misuse, alteration or abnormal conditions of operation or handling.

To obtain warranty service, contact a Fluke Service Center or send the product, with the description of the difficulty, postage prepaid, to the nearest Fluke Service Center. Fluke assumes no risk for damage in transit.

Fluke will, at our option, repair or replace the defective product free of charge. However, if we determine that the failure was caused by misuse, alteration, or abnormal condition of operation or handling, you will be billed for the repair. The repaired product will be returned to you, transportation prepaid.

THIS WARRANTY IS EXCLUSIVE AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR USE. FLUKE WILL NOT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OR LOSS WHETHER IN CONTRACT, TORT, OR OTHERWISE.

Contents

Section Title Page

Section Title Page

Section Title Page

Section Title Page

Section Title Page

Section Title Page

Appendices

Figures

Figure Title Page

Where Am I?

Getting Started		A description of the parts of the 9100A/9105A, what they do, how to connect them, and how to power up.
Automated Operations Manual		How to run pre-programmed test or troubleshooting procedures.
Technical User's Manual	***You Are Here***	How to use the 9100A/9105A keypad to test and troubleshoot your Unit Under Test (UUT).
Applications Manual		How to design test or troubleshooting procedures for your Unit Under Test (UUT).
Programmer's Manual		How to use the programming station with the 9100A to create automated test or troubleshooting procedures.
TL/1 Reference Manual		A description of all TL/1 commands arranged in alphabetical order for quick reference.

Section 1
Overview

The 9100A/9105A system is designed to test and troubleshoot digital electronic boards that are controlled by microprocessors. Testing precedes troubleshooting. If a board proves to be defective in a test, you can troubleshoot it to locate the fault. The board under test is called a UUT (Unit Under Test).

Testing and troubleshooting can be done in two ways:

- **Automated:** Using test or troubleshooting programs that are stored on a user disk. The *Automated Operations Manual* describes the system in enough detail for this purpose.
- **Keystroke:** The user typically employs a combination of automated procedures and manual keypad commands to test or troubleshoot a UUT. To do so,. the user should be familiar with both the UUT and the system.

This manual is primarily for the user who wishes to test a UUT in keystroke mode, using built-in tests and stimuli manually invoked at the operator's keypad. The manual also describes automated testing and troubleshooting. You should already have read *Getting Started*, which describes the basic parts of the 9100A and 9105A.

Among the topics covered in this manual are:

- **System Overview:** Section 2 provides an overview of the 9100A/9105A mainframe, pod, probe, and I/O modules. Approaches to testing and troubleshooting are also discussed.
- **Common 9100A/9105A Operations:** Section 3 describes how to use the operator's keypad and display for common procedures such as selecting a disk drive, executing the RUN UUT command to simulate normal UUT operation, or running an automated test. Also described is how to correct some common setup-type errors.
- **Self-Tests, Configuration, and Calibration:** Section 4 shows how to:

    - Perform system self-tests, which check that the pod, probe, and I/O modules are connected to and communicating with the system.

    - Configure the system by restoring saved system settings from the user disk.

    - Calibrate the system by restoring saved calibration data from a user disk or, if necessary, by recalibrating the system.

- **Keypad Reference:** Section 5 is a detailed alphabetic reference to all functions of all keys on the operator's keypad. The section contains some examples on the control and synchronization of input/output through the probe and I/O modules.
- **Test Techniques and Examples:** Section 6 contains examples of the use of built-in tests such as BUS, RAM, and ROM to quickly verify the integrity of basic UUT circuits.

    Examples show the use of the PROBE and I/O MOD commands and stimuli commands (such as READ, WRITE, and STIM) to test a node. Such tests are easily performed using the pod, probe, or I/O modules to exercise a node while comparing the node's responses to those from a known-good UUT.

- **General Maintenance:** Section 7 describes periodic maintenance and cleaning, voltage settings and fuses, shipping of your system should it require repair, and service support.
- **Glossary:** Section 8 contains definitions of commonly used terms.

The appendices in this manual cover:

- **Keypad-Syntax Quick Guide:** Appendix A contains an alphabetized list of all keypad syntax diagrams which appear in Section 5, "Keypad Reference."
- **I/O Module Clip/Pin Mapping:** Appendix B contains tables which indicate the correspondence between these sets of pins.
- **Pod-Related Information:** Appendix C provides a summary of the pod-specific information available in the *Supplemental Pod Information for 9100A/9105A Users Manual.*
- **Standard Equipment, Options, and Accessories:** Appendix D lists 9100A standard equipment, 9105A standard equipment, and options and accessories for both systems.
- **9100A/9105A Specifications:** Appendix E contains physical and electrical specifications for the 9100A/9105A and its test devices.
- **Fluke Sales Offices and Technical Service Centers:** Appendix F lists these worldwide locations.

An index is provided at the end of this manual.

Section 2
Introduction to the 9100A/9105A System

THE 9100A/9105A APPROACH TO TESTING AND TROUBLESHOOTING 2.1.

The 9100A/9105A has been designed to make the testing and troubleshooting of digital boards a structured process. Although you can use the system to test or troubleshoot UUTs that do not have microprocessors, such use under-utilizes the system's hardware and software tools.

Testing and troubleshooting should be structured according to the following sequence.

Gain control of the UUT by connecting the pod to the microprocessor bus.

Partition the UUT into major functional blocks. Then, using the pod, probe, or I/O module, run functional tests on the UUT. The tests write digital stimulus patterns to the UUT and gather the resulting responses from the UUT to identify any faults in its major functional blocks.

Use the built-in Guided Fault Isolation (GFI) program and stored stimulus programs to pinpoint the faulty component or node whose presence was indicated by the functional tests. If stimulus programs are not available, the operator's keypad and display can be used to execute and monitor stimuli.

During the troubleshooting process, responses that the stimuli generate must be compared to those expected from a known-good UUT and a good/bad verdict reached about the response. If the response is bad, GFI can either declare the tested component (or node) to be faulty, or can recommend another component (or node) to test in an effort to find the cause of the fault.

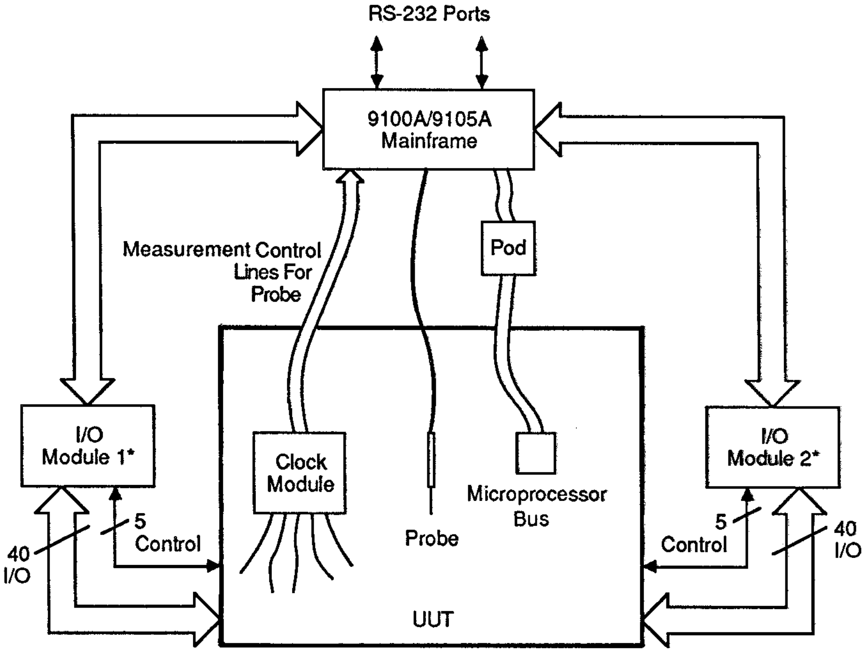
The pod can directly control or sense anything on the UUT that can be controlled or sensed by the UUT microprocessor. The probe and I/O module can stimulate and measure the UUT's digital circuitry, even that which is not accessible to the microprocessor.

Figure 2-1 is a functional overview of the 9100A/9105A system connected to a UUT. A stimulus or measurement through the probe or I/O modules can be clocked and synchronized by the pod's valid address or data cycles or by events accessed through external lines.

Solving Troubleshooting Problems 2.1.1.

An experienced person troubleshooting a UUT typically starts by learning about the fault and using the fault information to guess which functional block is defective. A functional test can be used to check the suspect functional block. If the guess is wrong, the process can be repeated until the faulty functional block is discovered. Subsequent troubleshooting of the functional block then narrows down the fault to a suitably low level, such as a single component.

The amount of information needed to locate the fault is a function of the fault itself and the troubleshooter's experience. The information varies between two extremes: a simple verbal description, or the result of probing all possible UUT nodes.



* Up to four I/O modules may be used.

Figure 2-1: 9100A/9105A System Overview

Troubleshooting Equipment Philosophy 2.1.2.

Most troubleshooting equipment is designed to replace experienced troubleshooters by less experienced ones using more hardware or software tools. A problem arises if novices are provided with tools but not with methods.

The absence of methods leads to inefficiency: for example, probing hundreds of nodes before knowing which functional block failed. Another problem is that available information and expertise are not used enough.

To solve these problems, the equipment should provide the troubleshooter with both the right tools and a structured method. In addition, the environment should allow the problem to be carefully structured. Lastly, procedures should build upon previous experience.

Testing and Troubleshooting with the 9100A/9105A 2.1.3.

The system has two basic functions:

- Testing: Locates the defective functional block using, for example, test programs or built-in tests.
- Troubleshooting: Traces the fault using (for example) GFI or manual CRC signature analysis.

Testing should precede troubleshooting; the system's design makes this easy. For example, running a program using EXEC is easier than collecting CRC signatures. In the 9100A/9105A, troubleshooting is done using:

- Manual or pre-programmed procedures: These can be used in any desired combination. This provides the greatest flexibility for experienced users.

- GFI: Troubleshooting is almost entirely pre-programmed, using the built-in GFI program. GFI can test a node and then suggest the next node to test.

9100A/9105A SYSTEM BLOCK DIAGRAM DESCRIPTION 2.2.

Figures 2-2 and 2-3 are the block diagrams of the 9100A and 9105A systems, respectively. The following are brief descriptions of all the blocks.

Main PCB: The main printed circuit board contains these major system components:

- Motorola 68000 microprocessor.
- Read/write (RAM) memory plug-in modules.
- Read-only (ROM) memory for the operating system.
- Floppy disk drive controller.
- Pod interface.
- Power supplies for the RS-232 ports, operator's display, and video.
- Ports for the plug-in video and SCSI/clock cards and other future expansions.

Video Interface: This plug-in card contains the video display interface circuits. The card may be configured for use with the Fluke monochrome monitor or an IBM-compatible color monitor. The video outputs are isolated from the system common because the monitors are earth-grounded. This card is used with the programmer station option in the 9100A or with the video output option in the 9105A.

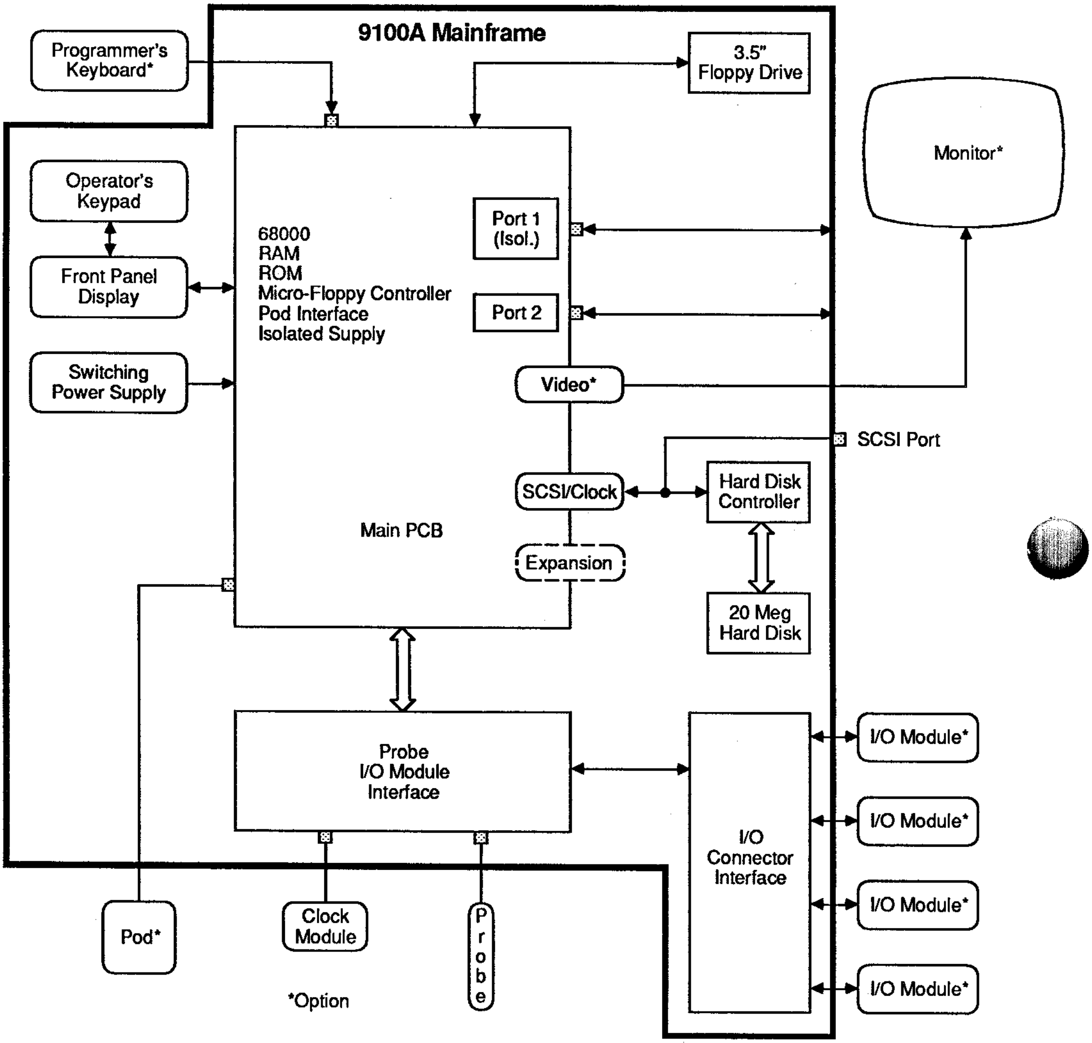


Figure 2-2: 9100A System Block Diagram

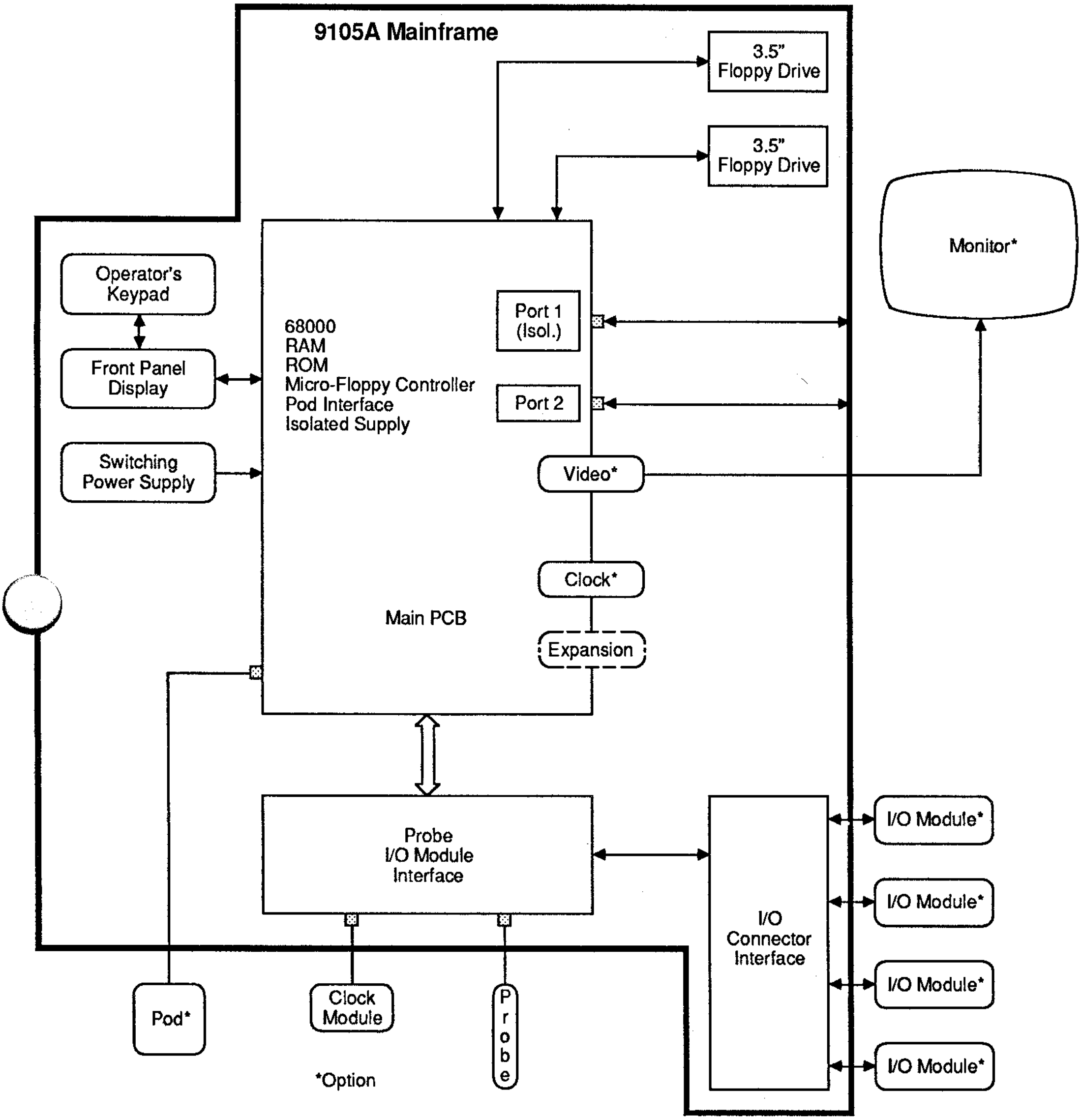


Figure 2-3: 9105A System Block Diagram

Monitor: Monochrome monitor for the programmer interface or for an extended user interface. The unit has 24 lines, each with up to 80 characters.

Programmer's Keyboard: The keyboard used during program development. It has a general purpose ASCII character set, editor functions, and configurable (softkey) functions.

SCSI/Clock Card (9100A only): This is the Small Computer Systems (SCSI) Interface to the hard disk system from the microprocessor. The SCSI port is available externally for future options. The card also contains the battery backed up real-time clock circuit, which is standard in the 9100A.

Clock (9105A only): An option card, containing a real-time clock with a battery backup.

Hard Disk Controller (9100A only): A Winchester hard disk controller card, which communicates with the hard disk using the industry standard ST506 interface. An interface to the host microprocessor is provided via the SCSI bus and the SCSI controller card.

Hard Disk (9100A only): A 20 Mbyte, 3.5-inch hard disk drive using the ST506 interface to the rest of the system. This drive contains system and user software.

Floppy Disk Drives: These drives accept 3.5-inch double-sided double-density micro-floppy disks for user software in the 9100A (one drive) or system and user software in the 9105A (two drives).

Operator's Keypad and Display: The user interface for testing and troubleshooting. A Z8 microcontroller system refreshes the 26 x 256 dot vacuum fluorescent display and scans the user keypad. Communication with the main system microprocessor is through a serial port.

Probe and I/O Module Interfaces: The interface between the microprocessor and the probe system, clock module, and I/O modules. The assembly contains the complete probe measurement and stimulus circuits, and the interface to the probe and its external clock module lines.

Clock Module: Input signal conditioning and buffering is provided for the external measurement control lines (START, STOP, CLOCK, and ENABLE) used by the probe system.

Probe: Input signal conditioning and output drive are provided. Also included are three logic level indicator lights.

I/O Connector Interface: This is primarily a data distribution card that provides connections for communication with four external I/O modules. It also has power supplies for the I/O module stimulus features.

I/O Modules: Each of these options is an integrated measurement and stimulus system for up to 40 I/O channels. Each channel has the measurement capability of the probe, at reduced speed. Each I/O module has external lines, used to synchronize data gathering to external events. A variety of adapters is available for connecting I/O modules to the UUT.

OVERVIEW OF THE 9100A/9105A MAINFRAME 2.3.

This section briefly describes the status lights, function keys, and softkeys on the operator's keypad and display. Visible features (such as connectors) on the mainframe side and rear panels are also described.

Figure 2-4 shows the operator's keypad and display, which serve as the user interface for running test programs, the GFI troubleshooting facility, and built-in functional tests and stimuli.

Commands and information are entered through the keypad and are monitored on the display, which also provides the user with instructions, responses, error messages and UUT fault messages.

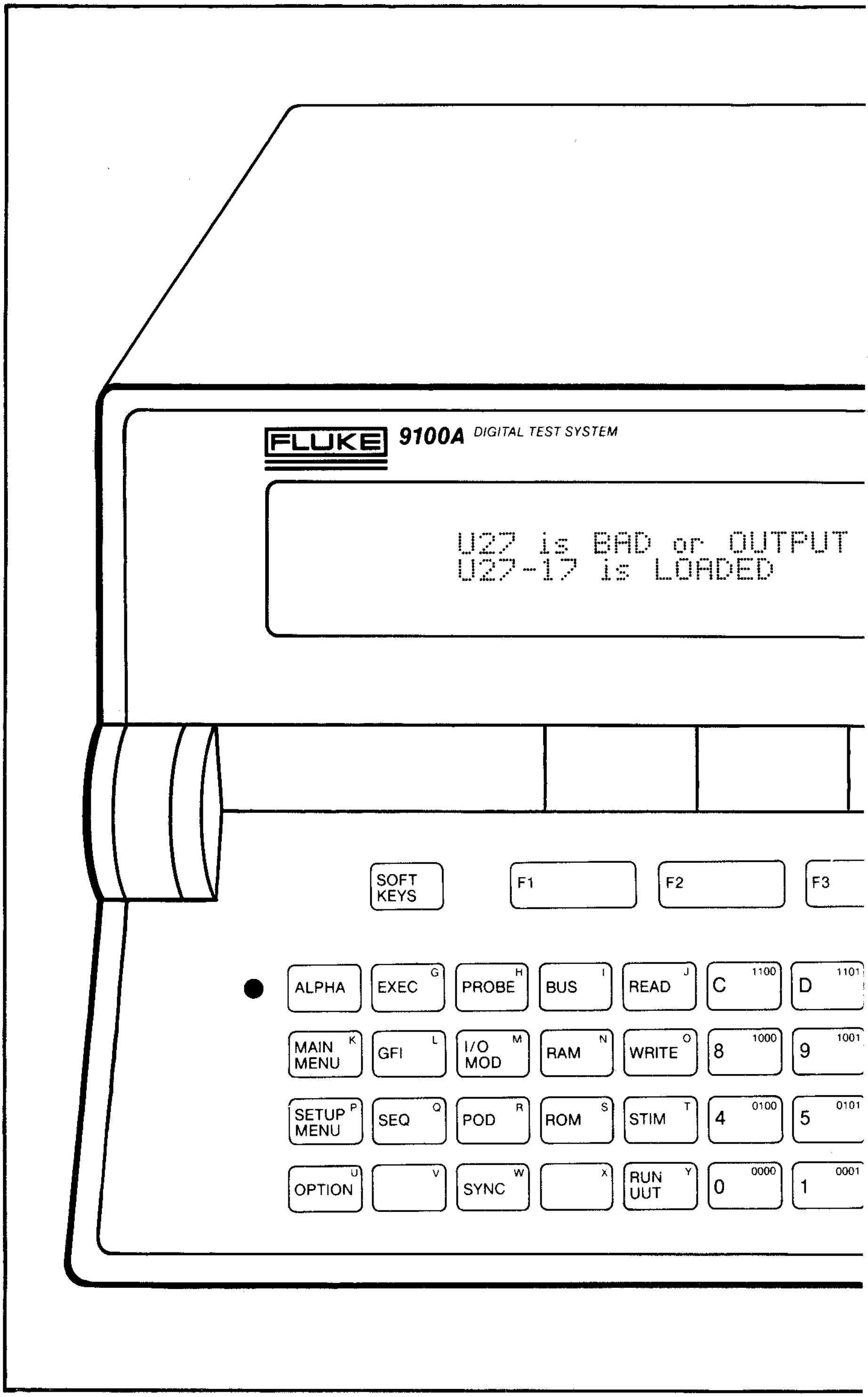


Figure 2-4: Operator's Keypad and Display

The Operator's Display 2.3.1.

This vacuum fluorescent display has three lines of up to 40 characters a line, with graphics capability. If there is no keypad or display activity for about 20 minutes, the display blanks; while it is blanked, the top two and bottom two status lights blink. No data is lost. Pressing any key on the keypad restores the display, and the key pressed is ignored.

Status Lights 2.3.2.

BUSY: Commands are being executed. If flashing, the system is looping (executing a command repetitively).

STOPPED: Command execution has halted.

RUN UUT: The pod microprocessor has control of the UUT and is executing instructions from UUT memory.

STORING SEQUENCE: The system is storing your current keystrokes as a numbered sequence for future use.

DISK ACCESS: A disk is being accessed or parked.

MORE SOFTKEYS: There are too many softkey labels to display. Press the SOFT KEYS key to scroll hidden labels into view.

MORE INFORMATION: More information is available on the display. Press an arrow key to see it.

ALPHA (next to the ALPHA key): The cursor is on a field which requires an alphanumeric entry. When the light is on, each function key represents an alphanumeric character shown in red on the upper right corner of the key.

The Operator's Keypad 2.3.3.

The operator's keypad and display, as the primary user interface, can be used to invoke nearly all functions that the system performs. Keys on the keypad can configure the system, run user software, execute built-in tests and stimuli, direct the built-in GFI troubleshooting facility, and manipulate files. Most of the function and numeric entry keys also serve as alphanumeric keys.

Section 5, "Keypad Reference," describes each key in detail.

Softkeys

F1 through F5: Each key, referred to as a softkey, selects the function indicated by the softkey label displayed directly above it. Softkeys are used to specify command fields, or to invoke seldom-used functions from a menu, or both.

SOFT KEYS: Scrolls hidden softkey labels into view if the MORE SOFTKEYS light is on.

Cursor Control

The left, right, up, and down arrow keys are used to move the cursor in the required direction. These keys are also used to scroll more information into view when the MORE INFORMATION light is on.

Numeric and Alphanumeric Data Entry

The white keys (0 through 9 and A through F) comprise a hexadecimal keypad. The keys can be used to enter data in decimal or hexadecimal notation into a command's numeric field if the cursor is at that field.

If the cursor is at an alphanumeric field, the ALPHA light comes on and the white keys A through F become alphanumeric. In this mode, the grey function keys are used for entry of the other alphanumeric characters, which are marked in red on the grey keys.

ALPHA: Changes the keypad from alphanumeric mode to command (function) mode and turns the ALPHA light off.

Command Entry

ENTER/YES: Serves as a YES response to questions on the display or enters the displayed command.

CLEAR/NO: Serves as a NO response to questions on the display. Also used to delete the last character typed into a field.

Control

REPEAT: Repeats the last command entered.

LOOP: Repetitively executes the last command entered.

STOP: Stops program or command execution.

CONT: Causes a program or built-in test to continue execution after stopping.

Menus

SETUP MENU: Sets system configuration variables such as communication port configurations and the real-time clock. These settings can be saved for future use along with some I/O MOD, MAIN MENU, PROBE, RAM, ROM, and SYNC settings.

MAIN MENU: Performs miscellaneous functions, including self-tests, calibration, and fault-reporting methods.

Pod Tests and Functions

BUS: Tests UUT power and the microprocessor bus.

RAM: Performs probabilistic (FAST) or deterministic (FULL) tests of the UUT RAM.

ROM: Tests the UUT ROM by comparing its CRC signatures to those saved from a known-good UUT ROM.

READ: Displays data read by the pod at specified memory, I/O, or other addresses.

WRITE: Writes data through the pod to specified memory, I/O, or other addresses.

STIM: A pattern of READs and WRITEs, specified by a RAMP, TOGGLE, or ROTATE function, used to raise a fault condition or to troubleshoot a known fault.

RUN UUT: Transfers control of the UUT to the pod microprocessor, which executes program instructions from UUT memory.

POD: Selects miscellaneous pod-dependent functions, such as quick pod-resident tests.

OPTION: Toggles the pod-dependent options display on and off, allowing options to be selected before command entry. Not all commands have options.

Probe and I/O Module Functions

PROBE: Causes outputs to and inputs from the UUT through the probe. Data is input or output at times set by the SYNC command.

I/O MOD: Causes outputs to and inputs from the UUT through the I/O modules. Data is input or output at command entry or at times set by the SYNC command.

SYNC: Specifies the source and synchronization of the clock signal used for inpuι or output through the probe and I/O modules.

Mode Control

EXEC: Executes a program from a given UUT directory.

EDIT: 9100A only. Transfers control of the system to the programmer's monitor and keyboard.

GFI: Runs the GFI (Guided Fault Isolation) algorithm to troubleshoot a UUT in GFI or UFI (unguided) mode.

SEQ: Allows the operator to store a numbered sequence of keystrokes that can be reproduced later by entering the stored number.

Miscellaneous

HELP: Turns the help window on and off, providing either information about a displayed fault condition or, if the ALPHA light is on, a listing of possible entries.

RESET: Resets hardware and all fields to power-up default values. GFI information is not lost.

Side Panel Connectors and Functions 2.3.4.

Figure 2-5 shows the 9100A and 9105A side panels with labels for the connectors and functions described below.

RESTART: When this button is pushed, the system resets and goes through its power-up sequence.

KEYBOARD: A connector for the programmer's keyboard.

POD: The pod should be plugged in here and secured by the connector's locking mechanism. The UUT power should be off and the 9100A/9105A power on when connecting to or disconnecting from a UUT.

COMMON: A common, non-earthed system reference.

CLOCK MODULE: A connector for the clock module.

CAL OUT: A signal output used, when calibrating the probe compensation, to monitor the calibration on an oscilloscope. COMP ADJ is trimmed until the best square wave possible is obtained at CAL OUT.

COMP ADJ: A trimmer capacitor used to adjust the probe compensation.

PROBE: A connector for the probe.

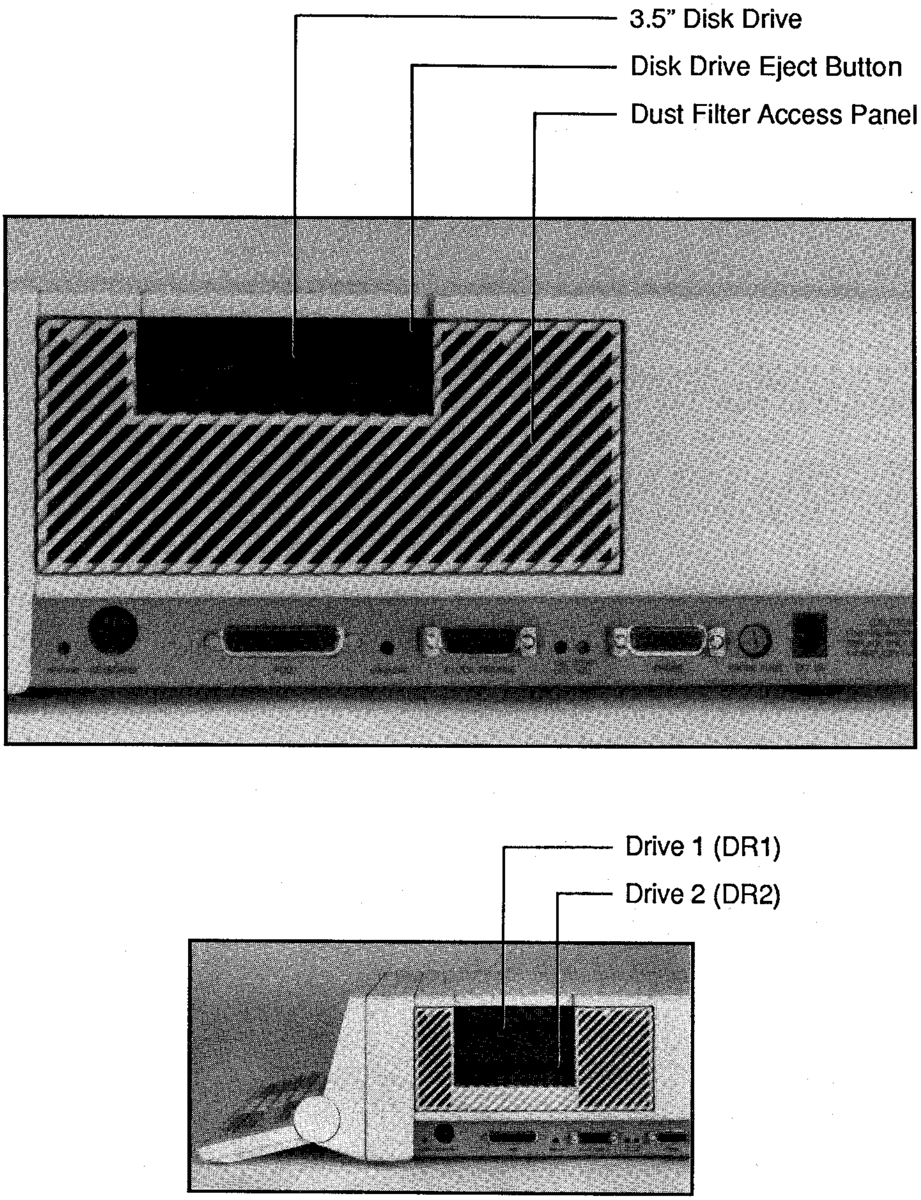
PROBE FUSE: 0.25 amp, 250 volt, fast blow type.

EXT SW: A telephone handset type connector for an external switch.

3.5-inch disk drive: Accepts double-sided double-density micro-floppy disks.

Disk drive eject button: Used to eject disks.

Dust filter access panel: This panel is removed to allow cleaning of the internal dust filter.



The 9105A, showing its different disk drive arrangement.

Figure 2-5: 9100A and 9105A Side Panels

Rear Panel Connectors and Functions 2.3.5.

Figure 2-6 shows the 9100A/9105A rear panel with labels for the connectors and functions described below.

I/O MODULES: Connectors are provided for up to four parallel I/O modules.

VIDEO: The monitor plugs in here.

TRIGGER OUTPUT: Can be connected to the "external trigger" input on an oscilloscope to observe events such as those comprising the READ and WRITE cycles.

MFI: A multi-function interface that uses the Small Computer Systems Interface (SCSI) standard to provide high speed access to mass storage devices like hard disks. On the 9105A, this slot is used for the real-time clock option. When the port is unused, keep its protective cover on.

EXPANSION: Plug-in cards for future system expansion can use this slot. The mainframe cover must be removed to gain access to the necessary connector on the main PCB.

PORT 1: RS-232 communication port. Pin 7 (Signal Ground) on the DB-25 connector goes to system common, which is **isolated** from ground. When the port is unused, keep its protective cover on.

PORT 2: RS-232 communication port. Pin 7 (Signal Ground) on the DB-25 connector goes to **earth ground**. When the port is unused, keep its protective cover on.

Fan air outlet: This outlet to the cooling fan should not be impeded when the mainframe is set up.

VOLTAGE: This switch selects the AC line voltage, 110 V or 220 V.

F1: This line voltage power fuse is a slow blow 2 A (110 volt) or 1 A (220 volt) type.

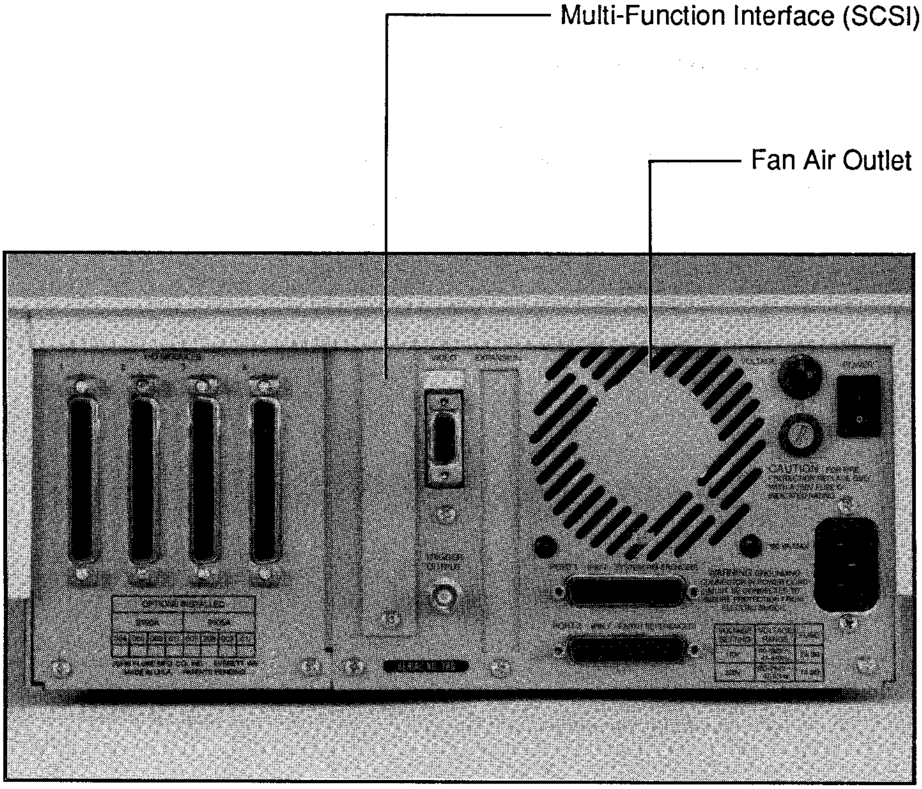


Figure 2-6: 9100A/9105A Rear Panel

POWER: This switch turns the mainframe power on or off.

POWER CONNECTOR: The power cord, which is connected here, must have a ground pin.

OVERVIEW OF PODS 2.4.

The Fluke interface pods let you use a 9100A/9105A to test or troubleshoot any circuit that is controlled by a microprocessor. The pod connects to the microprocessor bus and simulates the microprocessor in the UUT (Unit Under Test) and serves as an interface to the UUT bus (see Figure 2-7). The pod has two modes of operation:

- **Normal:** The pod adapts the architecture of the mainframe to the specific pin layout of the UUT microprocessor. This allows the mainframe to exercise all UUT buses while monitoring UUT activity. Typically, the pod executes READs or WRITEs, and the mainframe displays the results.
- **RUN UUT:** The pod microprocessor is connected via buffers to the UUT and serves as a substitute for the UUT microprocessor.

The mainframe supplies power to the pod. The UUT provides a clock signal to the pod, allowing it to function at the UUT's operating speed. Logic-level detection circuits on each line to the UUT detect any bus drive conflicts, or lines that are stuck high or low or are tied to each other.

Over-voltage protection circuits on each line to the UUT (except the clock line) guard against pod damage that could result from incorrectly connecting the pod to the UUT, or UUT faults that put potentially damaging voltages on lines to the microprocessor socket. A power level sensing circuit monitors the UUT power supply voltage. If UUT power rises above or drops below acceptable levels, the pod signals a bad power supply condition to the mainframe.

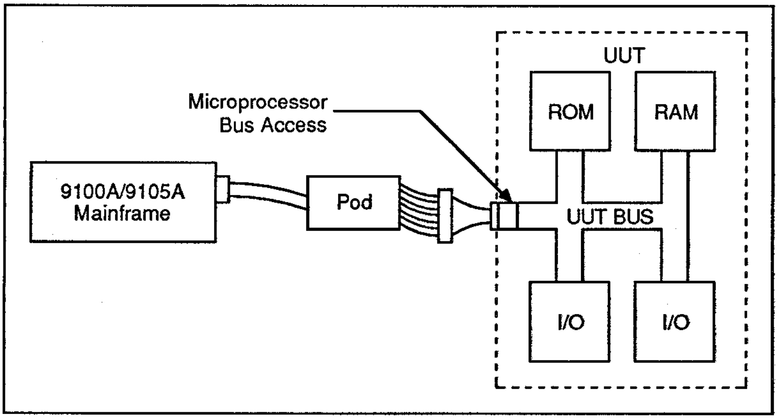


Figure 2-7: Interface to the UUT through the Pod

A self-test socket on the pod enables the mainframe to test pod operation. The connector normally attached to the UUT is secured to the pod self-test socket. While the self-test runs, the mainframe investigates the pod's internal functions.

OVERVIEW OF THE PROBE (AND CLOCK MODULE) 2.5.

The probe provides measurement (input) and stimulus (output) capability. Figure 2-8 shows a block diagram of the system formed by the probe and clock module.

Output 2.5.1.

Output is provided by the drive high and drive low circuits. Together, these blocks allow output to be pulsed high, low, or toggled between high and low. If the drive blocks are turned off, the output goes to a high-impedance state. Output can be synchronized to the freerun clock, the pod, or to external events using the clock module.

Input 2.5.2.

The probe contains a versatile set of measurement tools. The input signal is first processed by a set of input comparators, which measure the level (high, low, or invalid). At this point, both asynchronous and synchronous measurements can be made. Input threshold is adjustable to TTL, CMOS, or RS-232.

Asynchronous Measurements

Two different asynchronous measurements can be made. The 24-bit counter counts rising signal transitions, and the asynchronous level history block records the logic levels that are seen.

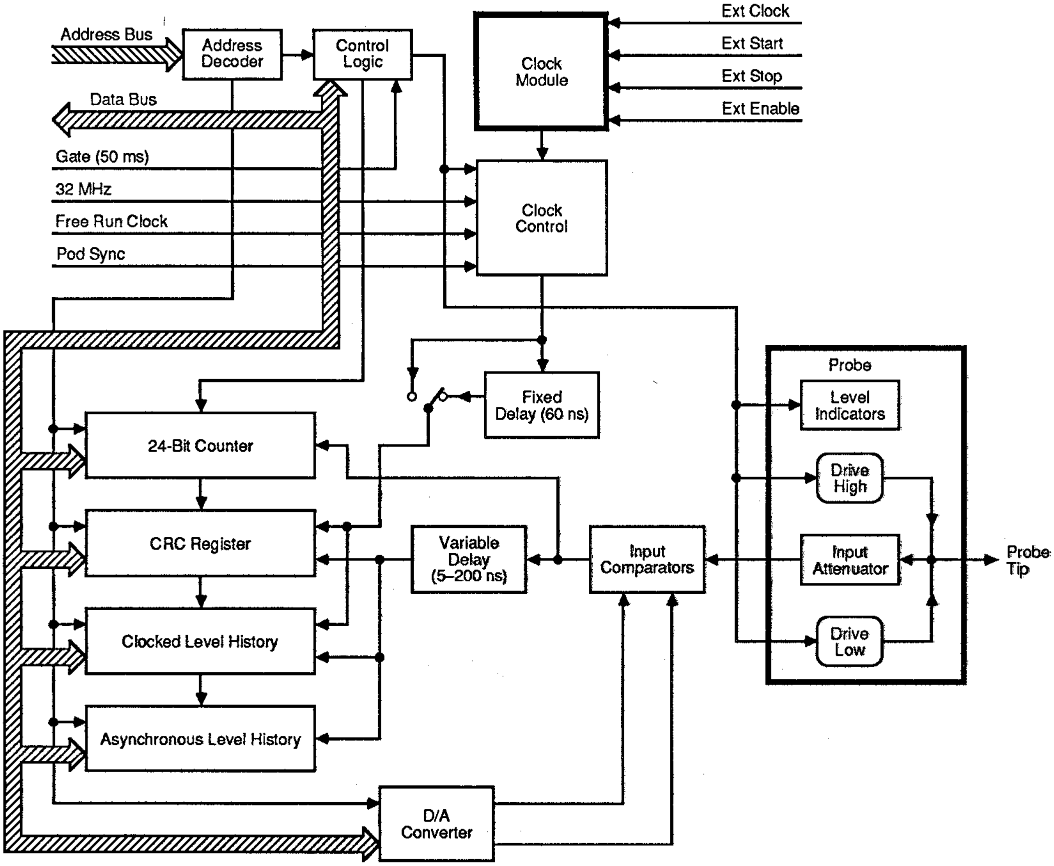


Figure 2-8: Probe and Clock Module System (Block Diagram)

Synchronous Measurements

On a valid clock edge, two different synchronous measurements can be made. A CRC signature is gathered in the CRC register, and the clocked level history block records the logic levels that are seen.

The fixed delay block is in series with the clock signal. The variable delay is in series with the input comparators and is calibrated to match its delay to the delay associated with the clock. This calibration helps ensure the accuracy of synchronous measurements.

Input Features

- CRC signatures: The probe input has a built in 16-bit register used to compute a CRC signature. The clock used in this computation can be qualified by the Start, Stop, and Enable lines. The pod's PodSync pulse can also provide this clock.
- Level history: Both the clocked level history register and the asynchronous level history register are used with the probe input. The clocked level history register records the logic levels present when a valid clock edge occurred. The asynchronous register acts similarly, except that it records level history independently of the clock and is therefore a useful glitch catcher.
- 24-bit transition counter: The probe has a 24-bit counter (24 bits and a latched overflow). When enabled, it counts rising signal transitions, from low (or invalid) to high levels.
- Frequency: The event counter can also be used to measure frequency. A 50 msec gate from the mainframe is used to enable the counter.

Level Indicators 2.5.3.

The probe has built-in lights to show the current asynchronous level information or to show the level information last seen by the synchronous level history latches. Red (high), green (low) and yellow (3-state or invalid) indicator lights are provided.

When the SYNC PROBE TO FREERUN CLOCK command is entered asynchronous information is displayed. In any other SYNC mode, the last clocked data is displayed.

Fuses 2.5.4.

Both the probe and clock module common lines are fuse protected. If either line is accidentally connected to a UUT power supply when a pod is also connected to the UUT, the appropriate fuse will blow. A message on the operator's display then specifies the fuse to replace.

OVERVIEW OF THE I/O MODULES AND ADAPTERS 2.6.

The I/O module is designed to provide measurement (input) and stimulus (output) capability on each of 40 channels. Figure 2-9 is a block diagram overview of one channel. A large variety of adapters are available for connecting I/O modules to different UUT components.

Output 2.6.1.

Output on each channel is provided by the drive high and drive low circuits. Together, these blocks allow the channels to be written high, low, or "off". If these blocks are turned off on a particular channel, the channel goes to a high-impedance state. The output can be latched or pulsed high, low, or 3-state.

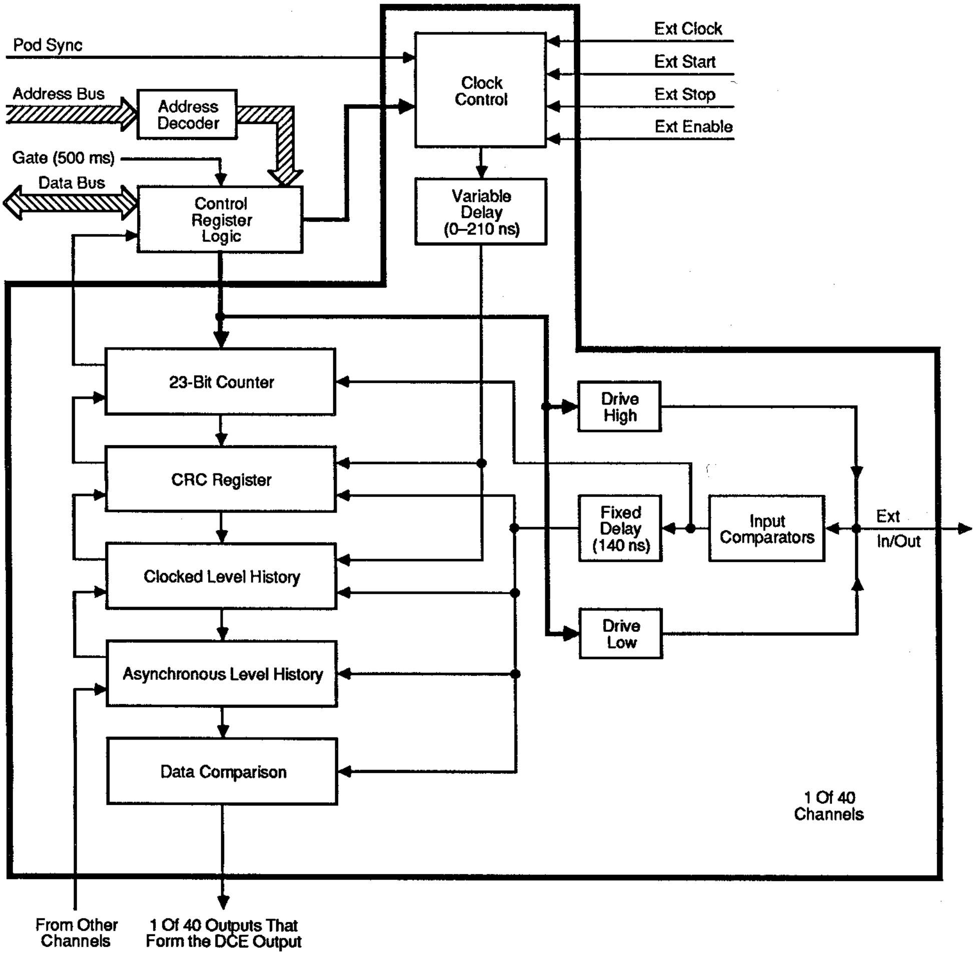


Figure 2-9: I/O Module Block Diagram

Output is not clocked, and is therefore asynchronous to the UUT. Pattern drive must be invoked by TL/1 program commands. All other output can be under either program or keypad control.

Input 2.6.2.

The I/O module contains a versatile set of measurement tools on each channel. The input signal is first processed by a set of input comparators which measure the level (high, low, or invalid). At this point, both asynchronous and synchronous measurements can be made.

Asynchronous Measurements

The signal from the comparators is continuously monitored by two different circuit blocks. Any rising signal transitions are counted by the 23-bit counter, and the signal is routed to one input of the 40-bit data comparison block. The asynchronous level history block records the logic levels seen.

Synchronous Measurements

On a valid clock edge, two different synchronous measurements can be made. A CRC signature is gathered in the CRC register, and the clocked level history block records the logic levels that are seen.

The fixed delay block is in series with the input comparators. The variable delay block is in series with the clock signal, and its delay is calibrated to match that associated with the clock. This calibration helps ensure the accuracy of synchronous measurements.

Input Features

- CRC signatures: Each I/O module line has a built in 16-bit signature register used to compute a CRC. The clock used in this computation can be qualified by the Start, Stop, and Enable lines. The PodSync pulse can also provide a clock.
- Level history: Both a clocked level history register and an asynchronous level history register are provided for each input. The clocked level history register will record the logic levels present when a valid clock edge occurred. The asynchronous register acts similarly, except that it records level history independent of the state of the clock. This register is a useful glitch-catcher.
- 23-bit transition counter: Each I/O module input has a 23-bit counter (23 bits plus a latched overflow). When enabled, it counts rising signal transitions from low (or invalid) to high level.
- Frequency: The event counter can also be used to measure frequency on each line. A 500 msec gate from the mainframe is used to enable the counter.
- Data comparison: All 40 I/O module data lines can be compared to a programmable 40-bit data register and qualified by a programmable 40-bit "don't care" register. When the input data equals the programmed value, an output is generated. This output is available at the DCE (data compare equal) pin on the side of the I/O module. DCE also generates an interrupt which can be handled by a TL/1 program. The comparison also can be programmed as a breakpoint for a pod in RUN UUT.

I/O Module Adapters 2.6.3.

I/O module adapters are used to connect an I/O module to components on the UUT. Clip modules and the flying lead module comprise two classes of I/O module adapters.

One or two adapters (depending on their size) can plug into an I/O module at the same time. Clip modules for 28- and 40-pin DIPs fit over both I/O module connectors "A" and "B"; other clip modules can fit over either connector "A" or "B" as desired. The part number of a clip module tells you the size of the DIP it clips over: a 28-pin clip module is labeled "Y9100A-28D."

Fuse 2.6.4.

The I/O module common line is fuse protected. If the line is accidentally connected to a UUT power supply when a pod is also connected to the UUT, the fuse will blow. A message on the operator's display then indicates that the fuse needs replacing.

9100A/9105A SOFTWARE OVERVIEW 2.7.

As shown in Figure 2-10, the 9100A software can be thought of as three main processes: an application shell, a run-time system, and programming support software. Only one of these processes can be active at any time. 9105A operation is very similar, except that it does not include the programming support software.

Application Shell Software 2.7.1.

At power-up, the 9100A/9105A enters the ROOT software module and loads the system software from the hard disk drive (9100A only) or a floppy disk drive (9100A/9105A) into system RAM. After performing various initializations, ROOT passes control to the application shell which sends a ready message to the operator's display.

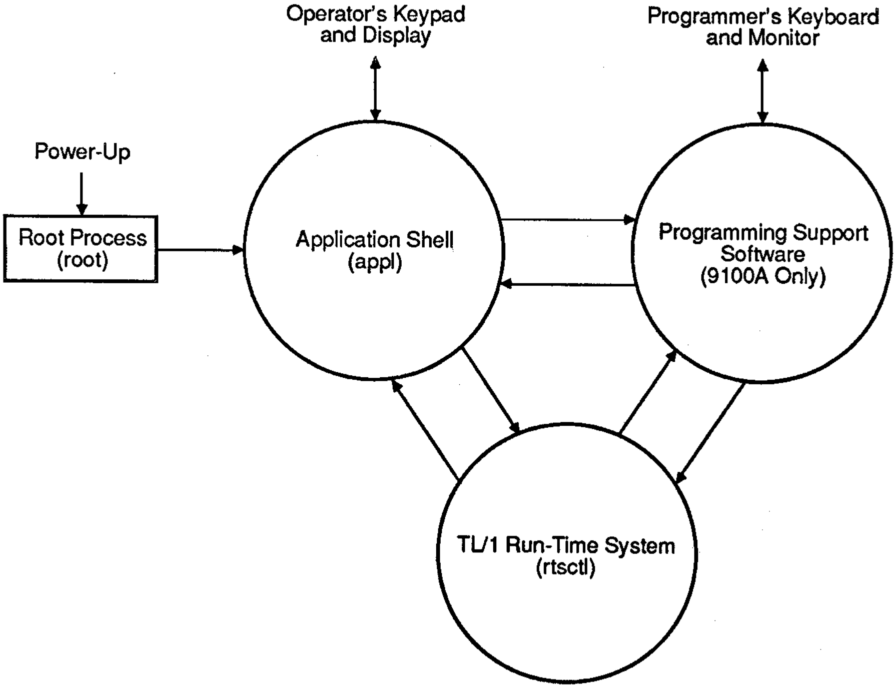

Operator's Keypad and Display
Programmer's Keyboard and Monitor
Power-Up
Root Process (root)
Application Shell (appl)
Programming Support Software (9100A Only)
TL/1 Run-Time System (rtsctl)

Figure 2-10: 9100A Software Overview

The application shell interacts with the user through the operator's keypad and display. Probe or I/O module calibration, self-tests (of the pod, probe, and I/O modules), and SETUP parameter changes are all handled by the application shell. All operator's keypad probe and I/O module operations are directly controlled by the application shell. In addition, the application shell can perform simple file operations such as formatting or copying a disk, removing text files from a disk, or printing text files from a disk.

Most frequently, the application shell will be used to execute TL/1 programs, guided fault isolation (GFI) procedures, and the built-in tests and stimuli that can be invoked at the operator's keypad.

TL/1 Run-Time System Software 2.7.2.

Pod operations (READ, WRITE, STIM, RUN UUT), built-in tests that use the pod (BUS, RAM, ROM), and TL/1 programs are executed by the run-time system under control of the application shell. When the run-time system is executed, the application shell waits for the run-time system to complete or stop.

Programming Support Software (9100A Only) 2.7.3.

Pressing the EDIT key on the operator's keypad causes the 9100A to enter the programming support software. This software interacts with the user through the programmer's interface: the monitor and programmer's keyboard. The programmer's support software includes an editor which is used to create or modify both TL/1 programs and the GFI database. The editor also includes utilities for formatting disks, copying files (or directories or disks), and printing out through the RS-232 ports.

The programming support software has a terminal emulator capability which lets the 9100A act as a terminal to communicate with another computer. The programming support software also includes a powerful facility for debugging TL/1 programs. To execute programs from the debugger, the programming support software executes the run-time system software in the same way the application shell does.

Simultaneously pressing the Shift and Quit keys on the programmer's keyboard causes a direct exit from the programming support software to the application shell.

Section 3
Typical 9100A/9105A Operations

This section shows you how to perform some typical 9100A/9105A operations. Most of the operations involve the use of the operator's keypad and display. You enter commands and information using the keypad and view responses on the display. The section describes how to:

- Power-up the 9100A/9105A and connect it to a UUT.
- Select a user disk drive.
- Format or copy disks.
- Execute an automated test.
- Simulate normal UUT operation using RUN UUT.
- Troubleshoot a UUT using UFI or GFI.
- Save or perform a keystroke sequence.

POWER-UP OF A 9100A 3.1.

To power-up the 9100A, you should:

1. Connect the probe, clock module, I/O modules, pod, monitor, and programmer's keyboard (as needed) to the 9100A/9105A.
2. Turn on the mainframe electrical power by setting the power switch on the back of the mainframe to the 1 (on) position. The system will proceed through its automatic power-up self-tests described below.
3. If you are using the monitor, turn the monitor's power switch to the 1 (on) position.
4. When the power-up procedure is complete, a ready message will appear on the operator's display.

During power-up, the 9100/9105A automatically performs built-in self-tests of the following internal systems:

- Operator's display.
- ROM.
- RAM.
- Pod interface.
- Communication port UART.
- EEPROM.
- Floppy disk controller.
- Probe and I/O module interfaces.

If any of the self-tests fail, a message to that effect appears on the operator's display. If the display will not function, the DISK ACCESS and RUN UUT lights will both flash. After any failure you are prompted with, "Press any key to continue". Doing so causes the system to continue with the next test. In some cases the system may be partially usable, even after a failure. Even if your system is usable, it is recommended that you contact a Fluke Service Center after a failure occurs.

POWER-UP OF A 9105A 3.2.

Power-up of a 9105A involves three actions: automatic self-testing, loading system software from Disk Drive 1 to RAM, and loading pod database information from Disk Drive 2.

Power-Up Procedure 3.2.1.

1. Turn on the mainframe electrical power by setting the power switch on the back of the mainframe (near the power cord) to the 1 (on) position. The system will proceed through its automatic power-up self-test.

2. After the self-tests are completed, you will be prompted to insert the disk labeled "System Disk 1" into Drive 1 (the upper micro-floppy disk drive).

3. After the loading of System Disk 1 is completed, you will be prompted to insert System Disk 2 into Drive 1.

4. When the loading of System Disk 2 is completed, a ready message will appear on the operator's display. (If no pod is connected, an additional message will appear on the operator's display to tell you so).

This completes the power-up procedure.

The 9105A automatically validates the software as it is loaded. The validation process checks that software is compatible with the 9105A, that system disks containing different software versions are not mixed together, and that the 9105A only runs authorized software. If you attempt to load System disks not authorized for your 9105A, loading is interrupted and the following error message appears:

```
This disk contains a serial number nnn
Please insert a system disk 1.
```

The serial number (nnn - shown in the previous error message) is the serial number of the inserted disk. You must insert the matching System Disk 1.

To use the 9105A for testing or troubleshooting, you will also need to do the following:

1. Connect the pod (for the microprocessor you will be using) to the 9105A mainframe.

2. Insert a User Disk appropriate for that microprocessor into Drive 2. This User Disk will be either Master User Disk 1, Master Disk 2, or a User Disk programmed at your organization for use with a particular UUT.

3. Press the RESET key to read the pod database from the User Disk you have inserted into Drive 2.

   If the following message appears on the operator's display:

   ```
   Pod name does not match data base name
   ```

   there is no User Disk in Drive 2 (or the wrong one has been inserted).

The correct User Disk should be left in Drive 2, since pod information on it is needed for some test or troubleshooting operations (pressing the RESET key, for example).

Master User Disk Contents 3.2.2.

Master UserDisk #1 contains pod database information for the following pods:

3548
3549
4050
8031
8041
8051
8051X
8080
8085
8086
8086MX
8088
8088MX
80186
80188
80286

Master UserDisk #2 contains pod database information for the following pods:

6800
6802
6809
6809E
68000
68000L
68000P
1802
6502
9900
Z80
Z8000
Z8001
Z8002

CONNECTING THE 9100A/9105A TO A UUT 3.3.

The procedure below assumes that the system is initially powered down and disconnected from the UUT. To connect the 9100A/9105A to the UUT:

1. Power up the 9100A/9105A.

2. Connect a pod, the same type as the UUT microprocessor, to the 9100A/9105A.

3. Attach the pod UUT connector to the pod self-test socket, and press the RESET key on the operator's keypad. This will ensure that the proper pod database is loaded into the 9100A/9105A.

4. Connect the pod UUT connector to the UUT and power-up the UUT.

5. Connect the probe, clock module, and I/O module common leads to UUT common points if these devices are to be used for testing.

6. Restore calibration data and the 9100A/9105A system configuration from disk as described in Section 4. Until this is done, the system cannot accurately test or troubleshoot a UUT.

AVOIDING CONNECTION PROBLEMS 3.4.

Your 9100A/9105A is smart enough to discover many types of errors that may occur in setting up a UUT for testing or troubleshooting. This section describes how to correct some common setup-type errors.

No Pod Connected 3.4.1.

If the pod is not connected to the 9100A/9105A, and you attempt an operation that needs the pod, the message "no pod connected" is displayed. To clear the error:

1. Switch off UUT power.

2. Make sure the pod is connected to the 9100A/9105A.

3. Connect the pod to the UUT.

4. Switch on UUT power and restart the test.

If the test fails again, the pod may be defective.

Bad UUT Power 3.4.2.

If the system detects that the UUT power supply is producing an incorrect voltage, the message "bad uut power" is displayed. To clear the error:

1. Switch off UUT power.
2. Make sure the pod is correctly connected to the UUT test-access socket. For details, refer to the pod manual.
3. Check power connections to the UUT.
4. Switch on UUT power and restart the test.

If the test fails again, the UUT power supply is probably defective.

Pod Timeout 3.4.3.

If the 9100A/9105A mainframe fails in an attempt to communicate with the pod, a pod timeout message will be displayed. If the mainframe can determine the cause, that information will follow the first message.

If the mainframe cannot determine the cause of failure, only the pod timeout message will be displayed. This failure is usually (but not always) caused by a defective UUT clock.

If, in the course of trying to determine the cause of a pod timeout, the timeout condition clears without the 9100A/9105A knowing why, a message "pod timeout recovered" is displayed.

USER DISK DRIVE SETUP 3.5.

The following steps show how to select a disk drive for a user disk.

1. **Insert your user disk** into the disk drive at the side of the system.

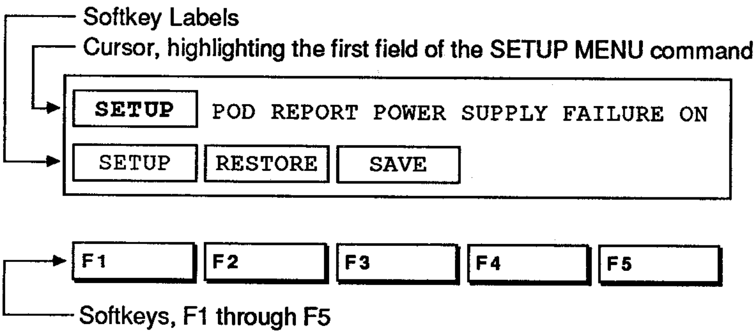
   Disk Drive 1 or disk Drive 2 may be used on a 9105A, but Drive 2 is recommended. (Drive 1 is used for system disks.)

   NOTE

   The master user disk (or a copy of it) is used when you do not have a user disk programmed for a particular UUT and you wish to troubleshoot in the "immediate" mode.

   If your system is a 9100A, the hard disk drive (HDR) is usually the user disk drive, but you may select Drive 1 if desired.

2. **Press the SETUP MENU key.** The SETUP MENU command appears on the display.



```
SETUP  POD REPORT POWER SUPPLY FAILURE ON
SETUP  RESTORE  SAVE
```

Softkey Labels
Cursor, highlighting the first field of the SETUP MENU command
F1
F2
F3
F4
F5
Softkeys, F1 through F5

Notice the cursor, the flashing block over the word SETUP. The cursor highlights a "field." A field is a word or number in a command that can be changed. Do not change any fields yet.

The words in boxes on the bottom line of the display are called "softkey labels." The keys F1 through F5 on the keypad are called "softkeys." To select a softkey function, press the softkey on the keypad just below the corresponding label.

NOTE

If, in step 2 when you pressed the SETUP MENU key, the display did not look like the one shown in the figure, you could make it do so by performing the following sequence:

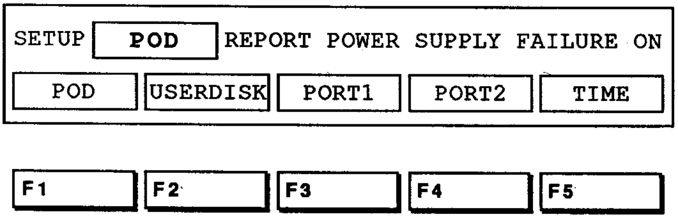
- *Press the left arrow key enough times to move the cursor all the way to the left.*
- *Press the F1 softkey (SETUP).*
- *Press the right arrow key once and then press the F1 softkey (POD).*
- *Press the right arrow key once and then press the F1 softkey (REPORT).*
- *Press the right arrow key once and then press the F1 softkey (POWER).*
- *Press the right arrow key once and press the F1 softkey (ON); then press the left arrow key enough times to move the cursor all the way to the left.*

What to do if you pressed the wrong function key: Once a function key is pressed, you can modify the function fields using the keypad. If you pressed the wrong function key, you will want to immediately abandon the function. This can be done quite easily.

Examine the ALPHA light on the left corner of the keypad. The light turns on whenever the cursor enters a field requiring an alphanumeric entry. If the ALPHA light is off, you can simply press the function key you wanted originally. The present function display will be discarded and the new function selections will be displayed. If the ALPHA light is on, press the ALPHA key to turn the light off and press the function key you wanted originally. When the light is on, the keys are interpreted as alphanumeric entries for the field where the cursor presently resides.

What to do if you pressed the wrong softkey: Press the right softkey and the old (wrong) one will be discarded.

3. **Press the RIGHT ARROW key.** The cursor moves to the next field (POD) and you see a new set of softkeys for that field. The MORE SOFTKEYS light comes on, reminding you that there are more softkey labels than those displayed.

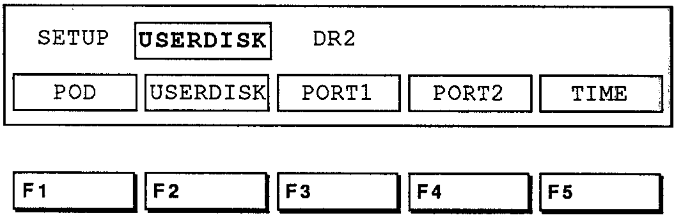


```
SETUP  POD  REPORT POWER SUPPLY FAILURE ON
```

4. **Press the F2 softkey,** which is below the USERDISK label. The field now contains the word USERDISK.

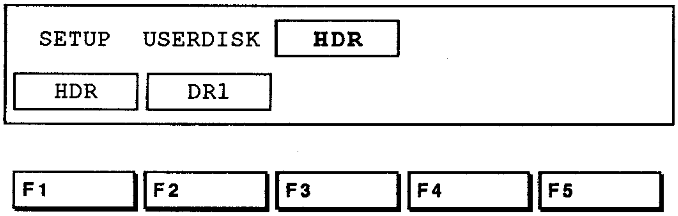
   If your system is a 9100A, the first line should read, "SETUP USERDISK HDR".

   If your system is a 9105A, the first line should look like the display shown below:



5. **Press the RIGHT ARROW key.** The display now shows which disk drives are available for use in the system

   The example below is from a 9100A.



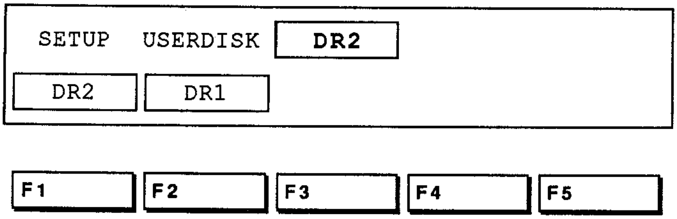
If you have a different system, your display will look different, as indicated below:

Your System	*Labels*	
9100A	HDR	DR1
9105A (two micro-floppy disk drives)	DR2	DR1

HDR is an internal hard disk drive.
DR1 (Drive 1) is the upper disk drive.
DR2 (Drive 2) is the lower disk drive.

6. **Press the softkey directly below the label of the disk drive you want to use.**

The example below is from a 9105A, so it shows only the two micro-floppy disk drives.



If DR2 is selected, the system will always execute programs from user disks in Drive 2 until told to do otherwise.

FORMATTING A DISK 3.6.

Before disks are used for the first time, they need to be formatted.

CAUTION

Formatting a disk that has been used before will erase all data stored on it.

To format a disk:

1. Power-up the system and insert the disk to be formatted in the upper drive, Drive 1.
2. Press the MAIN MENU key twice to display the command with the cursor at the first field. The red MORE SOFTKEYS light comes on because there are more softkey labels than those displayed.
3. Press the SOFT KEYS key to see additional labels.
4. Press the FORMAT softkey.
5. Move the cursor to the next field and press the DR1 softkey to specify Drive 1.
6. Press the ENTER key to format the disk in Drive 1.

The 9100A/9105A 3.5-inch disk drives use industry standard double-sided disks that can be obtained from Fluke or any commercial computer store. Each disk stores over 650 kilobytes of information.

To insert the disk into the drive, turn the disk so that the round metal hub is down and the arrow on the top left corner of the disk points toward the disk drive. Push the disk into the drive until it latches.

To remove the disk from the drive, press the black ejection button on the drive.

Do not expose the recording surface under the spring-loaded metal shield. Treat disks as you would treat audio cassettes. Do not expose them to:

- Moisture.
- Extreme temperatures.
- Direct sunlight.
- Magnetic fields.

A plastic locking tab is provided on each disk for write-protection. When a disk is write-protected, data can be read from (but not written to) the disk. To write-protect a disk, slide the tab over so that you can see through the window in the disk case. A write-protected disk cannot be formatted.

COPYING DISKS 3.7.

If you wish to make an exact copy of a disk, you must first format the disk that is to receive the copy. For details on the COPY command, refer to the MAIN MENU key in Section 5, "Keypad Reference."

Duplicating a Disk Using a 9100A/9105A 3.7.1.

This process creates an exact duplicate of the source disk. The destination disk will be overwritten. (See MAIN MENU COPY in the "Keypad Reference" section of this manual for more details.)

1. Place the source disk (the one you wish to copy) in Drive 1.

2. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

3. Press the SOFT KEYS key to display additional softkey labels, and then press the COPY softkey.

4. Press the right arrow key once and then press the DR1 softkey.

5. Press the right arrow key once and then press the DR1 softkey again. The display should now read:

   ```
   MAIN: COPY DISK FROM DR1 TO DR1
   ```

6. Press the ENTER key.

7. When prompted by the display, insert the source disk into Drive 1 and press ENTER. The BUSY status light will trun off when ready for the next step of the procedure.

8. When prompted by the display, insert the destination disk in Drive 1 and press ENTER. The BUSY status light will turn off when ready for the next step of the procedure.

Depending on the amount of available memory in the 9100A, steps 7 and 8 will be repeated until the source disk is completely copied to the destination disk.

When the copy is complete, the BUSY status light will turn off and the display will now read:

```
MAIN: COPY DISK FROM DR1 TO DR1 COMPLETE
```

System software disks can only be copied on authorized 9100A/9105A systems.

Disk Merge Copying Using a 9100A 3.7.2.

Merge copy will overwrite files of the same name but will leave all other files intact.

1. Place the source disk in Drive 1.

2. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

3. Press the SOFT KEYS key to display additional softkey labels, and then press the COPY softkey.

4. Press the right arrow key once and then press the DR1 softkey.

5. Press the right arrow key once and then press the HDR softkey. The display should now read:

```
MAIN: COPY DISK FROM DR1 TO HDR
```

6. Press the ENTER key. The BUSY light will turn off when the copy is complete.

Disk Merge Copying Using a 9105A 3.7.3.

Merge copy will overwrite files of the same name but will leave all other files intact.

1. Place the source disk (the one you wish to copy) in Drive 1.

2. Place the destination disk (the one to receive the copy) in Drive 2.

3. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

4. Press the SOFT KEYS key to display additional softkey labels, and then press the COPY softkey.

5. Press the right arrow key once and then press the DR1 softkey.

6. Press the right arrow key once and then press the DR2 softkey. The display should now read:

```
MAIN: COPY DISK FROM DR1 TO DR2
```

7. Press the ENTER key. The BUSY light will turn off when the copy is complete and the display will now read:

```
MAIN: COPY DISK FROM DR1 TO DR2 COMPLETE
```

System software disks can only be copied on authorized 9100A/9105A systems.

EXECUTING AUTOMATED TESTS 3.8.

The EXEC key executes automated functional test programs stored on a user disk. Its use involves pressing the EXEC key, entering a UUT directory name, entering a program name, and pressing ENTER. After you do this, the program controls testing.

The following procedure is a typical automated functional test of a UUT called ABC.

1. If you are using a 9105A, check to be sure that the user disk for testing your UUT is in drive 2.

2. Connect the pod to the UUT.

3. Make other connections to the UUT if instructed to do so.

4. Press the EXEC key and, if necessary, move the cursor to the left-most field. The display will now look like that below:

```
EXEC UUT ____ PROGRAM _____
```

NOTE

The fields may not always be blank, but they will change when you enter the names you want.

5. Enter a UUT directory name into the first field, using the operator's keypad. For this example, we will use a UUT named ABC. Now the display looks like this:

```
EXEC UUT ABC PROGRAM _____
```

NOTE

The field for the program name may not be blank, but it will change when you enter the name you want.

6. Move the cursor to the next field to the right and enter the program name using the operator's keypad. We will use TEST2 as the program name for this example. The display should now look like that below:

```
EXEC UUT ABC PROGRAM TEST2
```

7. Press the ENTER key to execute the program TEST2.

   If the UUT name and program name you enter are not found on the currently selected disk drive, the error message "PROGRAM DOESN'T EXIST ON DISK" appears on the display.

   If the UUT name and program name are found, the selected test is executed.

8. If a program has been written to ask you to enter values for the program, the operator's display will show each variable name and will place the cursor at the location where you are to enter the value.

9. Suppose this UUT has no faults. It would then pass the functional test. Your programmer might have designed a display message such as the following to tell you what to do next:

```
TEST2 PASSES.
PLEASE SET UP A NEW UUT FOR TESTING.
```

If, instead, the UUT is defective and the fault is detected by this test, the displayed message might be something like:

```
TEST2 FAILS.
USE GFI TO TROUBLESHOOT
```

RUN UUT TESTING 3.9.

Use the RUN UUT test to simulate normal UUT operation. The pod's microprocessor executes instructions found in UUT memory, starting at a specified address. Beyond lending the pod microprocessor to the UUT, the 9100A/9105A system does not interfere in the operation.

Starting this test involves pressing the RUN UUT key, the SPECIAL softkey, checking the default starting address, and pressing the ENTER key. If your pod allows breakpoints, you can use the BREAK softkey, before pressing ENTER, to specify a breakpoint.

The pod does not return control of the UUT to the 9100A/9105A unless a specified breakpoint address (exit) is encountered or the RUN UUT HALT command is entered while RUN UUT is in progress.

To execute the RUN UUT sequence:

1. Connect the pod to the UUT.

2. Press the RUN UUT key and use the left arrow key to move the cursor to the left-most field.

3. Press the SPECIAL softkey.

4. Move the cursor to the next field if you wish to change the starting address.

5. Move the cursor to the right-most field. If your pod allows breakpoint addresses, the following softkey choices will appear. Use one of them.

NOTE

If your pod does not allow breakpoints, you will not see the following softkey labels.

NO BREAK: This will cause the pod to start executing stored instructions at the address specified. The display should read something like:

```
RUN UUT SPECIAL ADDR FFFF0 NO BREAK
```

BREAK: Pressing this softkey allows you to type a UUT memory address at which the RUN UUT test is to stop.

The MORE INFORMATION light may come on if the command is so long that you have to use the right arrow key to view the next field.

6. Press the ENTER key to enter the command. You may also have to manually reset the UUT.

TROUBLESHOOTING WITH GFI 3.10.

The GFI (Guided Fault Isolation) key lets you use a powerful program—built into the 9100A/9105A system—to troubleshoot a defective UUT. GFI works by combining a set of basic troubleshooting rules with a database that describes your UUT. You need no special knowledge of either GFI or the UUT.

Running GFI involves pressing the GFI key and the RUN softkey. You then specify a UUT component name and component pin number and press the ENTER key. GFI starts running from the specified UUT location.

The procedure below shows a typical sequence you might use to isolate a fault on a defective UUT. This procedure was taken from a real UUT, but of course, you will have a different UUT and will probably even have a different fault, so the results you see will almost certainly be different. However, this example will be helpful in understanding what to expect when you run GFI to isolate faults on your UUTs.

Suppose you have a defective UUT and you wish to isolate the fault. To do so, a known-bad point on the defective UUT must be identified. The 9100A/9105A will use this as a starting point to begin backtracing. For your convenience, the system automatically selects the starting point if you have previously run an automated test. In such cases, when you press the GFI key, the name of the known-bad point is already entered into the GFI RUN command for you.

A typical GFI procedure is shown below:

1. Check to be sure the pod is connected to the UUT.

2. Check to be sure that any other connections required for your UUT have been made.

3. Test the UUT with an automated functional test that fails.

4. Press the GFI key and use the left arrow key to move the cursor to the left-most field.

5. Press the RUN softkey. At this point the operator's display will look something like that below:

   ```
   RUN GFI UUT DEMO REF U27 PIN 1
   ```

   If the functional test at step 3 left a hint, the REF and PIN fields are automatically filled in as default values. If there is no hint, you must specify the starting point.

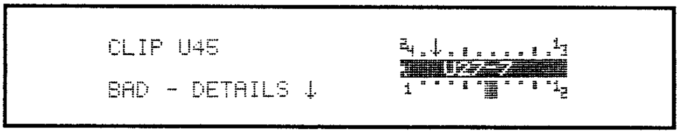
If for some reason you do not like this starting point, you can press the SUGGEST softkey and then the ENTER key to look at other recommended starting points for GFI. Then you can press the GFI key followed by the RUN softkey and enter another GFI starting point using the cursor control keys and the operator's display.

6. When you are ready to begin fault isolation at the point in the RUN GFI command, press the ENTER key. If the GFI database exists for the selected UUT, GFI takes over and starts backtracing from the selected starting point on the UUT. For this example, the display now becomes:

```
RUN GFI UUT DEMO REF U27 PIN 1
CLIP U27
PRESS BUTTON ON I/O MOD WHEN READY
```

7. Attach a clip over the component specified (U27 in this case) and press the ready button on the clip module used.

8. GFI tests and displays the condition of all pins of U27 and recommends the next component to test. A typical display is shown below:



```
CLIP U45
BAD - DETAILS ↓
```

The arrows on the displayed IC pins indicate bad input, output, or bidirectional pins. The words in the lower left corner display the status of each pin as the cursor is moved from pin to pin with the right arrow and left arrow keys.

The words in the upper left corner show the GFI recommendation for the next component to check. In this example, GFI decided U45 should be tested next.

9. It is usually appropriate to use the GFI recommendation for the next component to test, rather than to check the status of all the pins on the displayed IC. So, you would clip over U45 and press the clip module ready button.

   In this example, suppose that GFI decides that it has already isolated the problem and the display shows:

   ```
   OPEN CIRCUIT BETWEEN
   U45-3 AND U27-22
   ```

   GFI has identified the probable defect and will therefore stop execution.

10. In some cases you may decide to view the GFI summary or the suggestion list, but it is usually appropriate at this point to clear the GFI summary and the suggestion list. This is done by the following steps:

    Press the GFI key.

    Use the cursor control keys to move the cursor to the left-most field.

    Press the CLEAR softkey and then the ENTER key.

TROUBLESHOOTING WITH UFI 3.11.

UFI (Unguided Fault Isolation) is designed for a situation where the user wishes to use GFI's pin testing capability but does not need probing suggestions. In UFI troubleshooting the user may, for example, use a combination of test programs, built-in tests, and UFI. The UFI user is normally someone experienced who is familiar with the UUT, has a good idea why it failed, and can save time accordingly.

UFI is invoked with the GFI key after a functional test has been run. UFI tests only output pins and does not make probing recommendations. In place of the recommendations that GFI displays, the message "UNGUIDED MODE" will be displayed. A typical UFI procedure is shown below:

1. Check to be sure the pod is connected to the UUT.

2. Check to be sure that any other connections required for your UUT have been made.

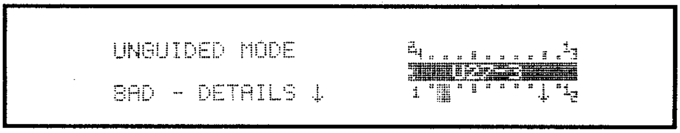
3. Test the UUT with an automated functional test. If the test fails, any hints (suggested starting points) that it generates are ignored by UFI.

4. Press the GFI key and use the left arrow key to move the cursor to the left-most field.

5. Press the RUN softkey. At this point the operator's display should be something like:

   ```
   RUN GFI UUT DEMO REF U27 PIN 1
   ```

6. When the ENTER key is pressed, each output pin on the IC is tested and the results displayed as for GFI:



UNGUIDED MODE

BAD - DETAILS ↓

7. If the user now decides to test a different IC or another pin on the same one, steps 4 through 6 are repeated. Step 3 can also be repeated with a different test program, if desired.

   UFI (like GFI) maintains a summary. There is no UFI suggestion list because UFI does not make probing suggestions. UFI (unlike GFI) does not accuse ICs or nodes that are apparently bad. This decision is left to the user's judgement.

USING KEYSTROKE SEQUENCES 3.12.

When developing UUT stimulus and test programs or when testing UUTs in the keystroke mode, it is frequently necessary to repeat a group of keystrokes on the operator's keypad of the 9100A/9105A. Rather than manually entering the sequence each time, it can be saved as it is entered and stored as a keystroke sequence. The sequence can be saved on the current userdisk or in a specified UUT directory for use at a later time.

For example, you might want to set up an 8255 PIA to output on port A and to verify that the data appeared correctly at an I/O module that is clipped to the 8255 chip. To read port A of the 8255 PIA requires mapping of 40 clip module pins to the I/O module pins to form an eight bit word. Since the 40-pin clip module pins map one-to-one with the I/O module pins (see Appendix B), you should set up word 1 of I/O module 1 as pins 37, 38, 39, 40, 1, 2, 3 and 4. These pins correspond to the pins of port A on the 8255 PIA.

Setting up the I/O module word involves many keystrokes and would be a good situation in which to use a stored keystroke sequence.

Storing A Keystroke Sequence 3.12.1

The following keystroke sequence will set up the necessary I/O module word for the example above and will save it as keystroke sequence 1:

SEQ
F2 (BEGIN)
—>
1
ENTER (the STORING SEQ LED will light)
I/O MOD
F5 (SET)
—>
F1 (1)
—>
F2 (WORD)
—>
F1 (1)
—>
3
7
—>
3
8
—>
3
9
—>
4
0
—>
1
—>
2
—>
3
—>
4
—>
0

```
ENTER
SEQ
F3 (QUIT)
ENTER
```

Performing a Keystroke Sequence 3.12.2

The following procedure uses the keystroke sequence defined above for I/O module 1 to read Port A of the 8255 PIA chip. The procedure assumes that an 80286 Pod is being used.

1. Use the SEQ key to set the I/O Module word by executing the previously saved keystroke sequence (sequence #1):

   ```
   SEQ
   F1 (PERFORM)
   —>
   1
   ENTER
   ```

2. Use the WRITE key with the following command to set the 8255 PIA control register for all ports:

   ```
   WRITE DATA 80 TO ADDR 4006
   ... (ADDR OPTION: I/O BYTE)
   ```

3. Use the WRITE key with the following command to write hexadecimal AA to port A of the 8255 PIA:

   ```
   WRITE DATA AA TO ADDR 4000
   ... (ADDR OPTION: I/O BYTE)
   ```

4. Use the I/O MOD key with the following command to read the I/O Module 1 data word:

   ```
   INPUT I/O MOD 1 WORD 1
   ```

The word that is read is added to the display as shown below:

```
INPUT I/O MOD 1 WORD 1 = AA
```

SUMMARY OF FREQUENT OPERATIONS 3.13.

Figure 3-1 is a flowchart of frequent operations. After power-up and self-tests, restore the system configuration stored on the user disk. Then restore saved calibration data from the disk or, if necessary, calibrate the system.

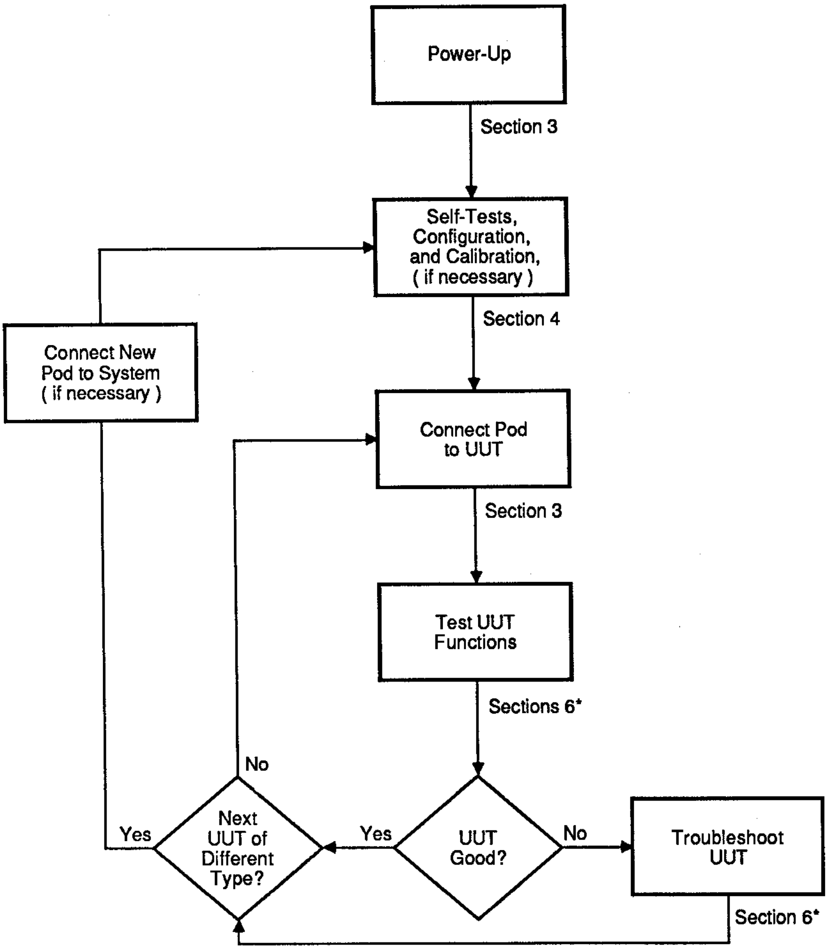
After configuration and calibration, ensure that the userdisk is in the correct drive. Use mode selection to specify whether or not the system stops when UUT faults are detected.

You are now ready to test or troubleshoot a UUT.

If parts of your system are changed, it should be recalibrated. The new calibration data can be stored on the old disk, writing over existing data.

If you change to a different type of UUT, you may need another disk for that UUT. Restore the system configuration for the new UUT from the appropriate user disk. In addition, the new system will have to be recalibrated, if the new UUT uses a different pod. The new calibration data can be stored on the UUT disk, writing over the calibration data on that disk.

Self-tests should be performed on pods, probes, and I/O modules, when they are first installed. If you change or swap any of these devices (for example, if changing from one type of UUT to another), self-tests should be performed on the newly installed device.


Power-Up
Section 3
Self-Tests, Configuration, and Calibration, (if necessary)
Section 4
Connect New Pod to System (if necessary)
Connect Pod to UUT
Section 3
Test UUT Functions
Sections 6*
No
Yes
Next UUT of Different Type?
Yes
UUT Good?
No
Troubleshoot UUT
Section 6*

* Also see the *Applications Manual*

Figure 3-1: Typical Test and Troubleshooting Procedures

INSTALLING NEW VERSION 9100A/9105A SOFTWARE 3.14.

The procedure for software installation depends on the type of mainframe. Refer the the appropriate procedure for the type of mainframe you are operating.

9100A Software Installation Procedure 3.14.1.

The following procedure should be followed to install a new version of the operating software for the 9100A.

WARNING

DO NOT RESTART OR TURN THE 9100A POWER OFF UNTIL ALL NEW SYSTEMS DISKS HAVE BEEN COPIED!

1. Copy the new System Disks and Programmer's System Disk (if you have the Programmer's Option) to the 9100A hard disk by selecting the command:

   ```
   MAIN: COPY DISK FROM DR1 TO HDR
   ```

   The first time you load new System disks from the factory, the 9100A writes validation data on them. The validation data allows the software to be loaded. If you receive the message:

   ```
   Please write enable system disk
   ```

   push the write-enable tabs on the new System Disks to the closed (write enable) position before loading. The System Disks are now valid, and cannot be loaded or copied on any other machine.

Write the serial number of the 9100A (located on the rear panel) on each of the disks to avoid mixing them up with disks from other systems.

2. Restart your 9100A by cycling the power or by pressing the RESTART button on the side panel. The 9100A should boot up and display the new software version number.

WARNING

ANY PROGRAMS THAT HAVE BEEN MODIFIED ON YOUR 9100A HARD DISK THAT HAVE THE SAME NAME AS THOSE ON THE MASTER USER DISKS WILL BE OVERWRITTEN!

3. If your upgrade includes new Master User Disks, copy them to the 9100A hard disk using the command:

```
MAIN: COPY DISK FROM DR1 TO HDR
```

9105A Software Installation Procedure 3.14.2.

The following procedure should be followed to install a new version of the operating software for the 9105A:

1. Cycle the 9105A power off then on.

2. Use the new System Disks to boot the 9105A.

   The first time you boot from the new System Disks, the 9105A writes validation data on them. The validation data allows the software to be loaded. If you receive the message:

```
Please write enable system disk
```

push the write-enable tabs on the new System Disks to the closed (write enable) position before loading. The System Disks are now valid, and cannot be copied or used to boot on any other machine.

3. Make a working copy of each of the System Disks. The original should be put aside as backups. To copy the disks follow the procedure in "Duplicating a Disk Using a 9100A/9105A".

Section 4
Self-Tests, Configuration, and Calibration

INTRODUCTION 4.1.

Every time you power-up, reset the system, or change parts of it, you should perform the following operations before using the system to test or troubleshoot a UUT.

- **Self-tests:** Ensure that the pod, probe, and I/O modules are connected and working correctly. The tests check that the devices are operating but the tests do not check the accuracy of the devices' measurements.
- **Configuration:** Specifies the disk drive to be used, port configuration, error reporting, and other system settings. Settings can be saved on and restored from the user disk.
- **Calibration:** Adjusts the system to measure data as accurately as possible.

An alternative to recalibration on power-up or system reconfiguration is saving calibration data (for that configuration) on a user disk and restoring it when necessary.

In addition to storing test programs and troubleshooting routines, a user disk can also store:

- Calibration data generated by calibrations.
- System configurations, including SETUP MENU command parameters and information used by the I/O MOD, PROBE, RAM, ROM, and SYNC commands.

If you reset the system or change any of its parts such as the pod, probe, or I/O modules:

- The self-tests should be repeated and the system must be recalibrated. New calibration data can be saved on the user disk, writing over the old data.
- Test and troubleshooting programs and the system configuration stored on user disks remain valid.

If you change to a different type of UUT:

- If the new UUT needs a new pod or I/O modules, the self-tests should be repeated and the system recalibrated. You can save new calibration data on the UUT user disk, writing over the old data.
- You may need another user disk for the new UUT. The system should be configured for the new UUT by restoring the configuration stored on the new UUT userdisk.

CAUTION

Once information for a certain type of UUT is stored on a user disk, your system should use only that disk for all UUTs of that type because the calibration data stored is unique to each type of UUT.

SELF-TESTS 4.2.

Three self-tests should be performed after you power-up the system or change the I/O module, pod, or probe. Before running the tests, check that the devices are connected correctly.

Probe Self-Test 4.2.1.

The probe self-test is a pass/fail check that the probe is connected to and communicating with the system. To execute this test:

1. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.
2. Press the SELFTEST softkey.
3. Move the cursor to the next field and press the PROBE softkey. The display should read:

   ```
   MAIN: SELFTEST PROBE
   ```

4. Press the ENTER key to perform the self-test. If the self-test fails, a failure message will be displayed. If this happens, check the probe connection and repeat the test carefully following any instructions. If the test fails again, the probe requires service.

Pod Self-Test 4.2.2.

The pod self-test comprehensively tests the pod. If it fails, an error message is displayed to explain the cause of failure. If you do not understand the message, refer to the pod manual or to the *Supplementary Pod Information for 9100A/9105A Users* manual. To run this test:

1. Make sure the pod is attached securely to the mainframe. Insert the pod UUT connector into the pod self-test socket and lock it. Refer to the pod manual and *Getting Started* for details.

2. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

3. Press the SELFTEST softkey.

4. Move the cursor to the next field and press the POD softkey. The display should read:

```
MAIN: SELFTEST POD
```

5. Press the ENTER key to perform the self-test. If the self-test fails, a failure message will be displayed. If this happens, check the pod connection and repeat the test carefully following any instructions. If the test fails again, the pod requires service.

6. If the self-test passes, remove the pod's UUT connector from the self-test socket and connect it to the UUT, which should be powered down. If necessary, refer to the pod manual for connection details.

I/O Module Self-Test 4.2.3.

The I/O module self-test is a pass/fail check that the module concerned is connected to and communicating with the system. To execute the test:

1. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

2. Press the SELFTEST softkey.

3. Move the cursor to the next field and press the I/O MOD softkey.

4. Move the cursor to the next field and identify the module to be tested with a number from 1 to 4. For example, if I/O module 2 is to be tested, the display should read:

```
MAIN: SELFTEST I/O MOD 2
```

5. Press the ENTER key. If the self-test fails, a failure message will be displayed. If this happens, check the I/O module connection and repeat the test, carefully following any instructions. If the test fails again, the I/O module requires service.

CONFIGURATION 4.3.

Configure the system each time you power-up or reset it or change parts of it. After doing so for one type of UUT, if you change to another, you should reconfigure the system. Configuration sets system parameters such as the disk drive to be used, error reporting, and port configurations.

Disk Drive Selection 4.3.1.

Steps 2 through 6 in Section 3, "Typical 9100A/9105A Operations," showed how to select a disk drive. This is only necessary if your user disk is not in the default drive. The default drives are:

- HDR (hard disk) in the 9100A.
- DR2 (Drive 2) in the 9105A.

After configuring the system as described in the following sections, you should verify that the UUT directory is, in fact, on the user disk in the drive specified by the configuration. To verify the drive:

1. Press the SETUP MENU key and use the left arrow key to move the cursor to the left-most field.

2. Press the SETUP softkey.

3. Move the cursor to the next field and press the USERDISK softkey.

4. Verify that the drive specified in the right-most field contains the user disk with the UUT directory you need. Possible drives are:

   HDR (internal hard disk): 9100A only.
   DR1 (Drive 1): Upper disk drive (9100A or 9105A).
   DR2 (Drive 2): 9105A lower disk drive.

Saving a System Configuration 4.3.2.

Saving a system configuration completely writes over the existing one on a user disk. You should not save a configuration on user disk unless authorized to do so.

System settings and other information specified by the following commands are saved as a system configuration:

- SETUP MENU (all commands).
- I/O MOD (SET).
- PROBE (SET).
- RAM (DEFINE).
- ROM (GET SIG).
- SYNC (all commands).
- SEQ (numbered stored sequences).

To save a system configuration:

1. Press the SETUP MENU key and use the left arrow key to move the cursor to the left-most field.

2. Press the SAVE softkey.

3. Move the cursor to the next field and press the SYSTEM softkey.

4. Move the cursor to the next field and press one of the following softkeys:

   USERDISK: Used if the user disk does not contain a UUT directory. The resulting display should read:

   ```
   SAVE SYSTEM SETTINGS IN USERDISK
   ```

   UUT FILE: Used if the user disk contains a directory for the UUT.

5. If you pressed the UUT FILE softkey in step 4, type the UUT directory name: for example, type "DEMO". The display should read:

   ```
   SAVE SYSTEM SETTINGS IN UUT FILE DEMO
   ```

6. Press the ENTER key to save all current system settings as a configuration.

Restoring a System Configuration 4.3.3.

To restore a system configuration (all system settings) from a user disk:

1. Press the SETUP MENU key and use the left arrow key to move the cursor to the left-most field.

2. Press the RESTORE softkey.

3. Move the cursor to the next field and press the SYSTEM softkey.

4. Move the cursor to the next field and press one of the following softkeys:

   USERDISK: If the system configuration was saved in the USERDISK directory. The resulting display should read:

   ```
   RESTORE SYSTEM SETTINGS FROM USERDISK
   ```

   UUT FILE: If the system configuration was saved in a UUT directory.

5. If you pressed the UUT FILE softkey in step 6, type the UUT directory name: for example, type "DEMO". The display should read:

   ```
   RESTORE SYSTEM SETTINGS FROM UUT FILE DEMO
   ```

6. Press the ENTER key to configure the system.

CALIBRATION 4.4.

Calibration is the process of adjusting the 9100A/9105A system so that individual variations in the time delays associated with the probe and pod are known and compensated for.

Five calibrations should be routinely performed on the system:

- Probe compensation calibration.
- Probe to external clock module calibration.
- Probe to pod calibration.
- I/O module external calibration.
- I/O module to pod calibration.

The calibrations listed above are of two types:

- **Compensation:** Matches impedances of two parts.
- **Data Against Clock Delay:** Ensures that data arrives at the receiving hardware at the same time as the signal clocking the data.

Calibration should be done when the system is first set up and at regular intervals (at least monthly) after that. Calibration is also necessary whenever devices attached to the system are changed or repaired. Probe compensation remains stable and is only necessary when a probe is first connected to the mainframe.

The section on "An Alternatives to Calibration," describes how to calibrate the system by restoring data generated by previous calibrations. This should be done after each power-up or reset before testing or troubleshooting a UUT.

Probe Compensation 4.4.1.

This calibration compensates the probe by matching its impedance to that of the cable connecting it to the system. Probe impedance is adjusted by "COMP ADJ" a trimmer capacitor located on the side of the system.

Figure 4-1 shows how an oscilloscope should be connected during the calibration. To compensate the probe:

1. Ensure that your oscilloscope and its probe are properly calibrated.

2. Hook the oscilloscope probe tip to the system's CAL OUT post. Hook the oscilloscope probe common clip to the system's COMMON post. The posts are located in labeled holes on the mainframe's side.

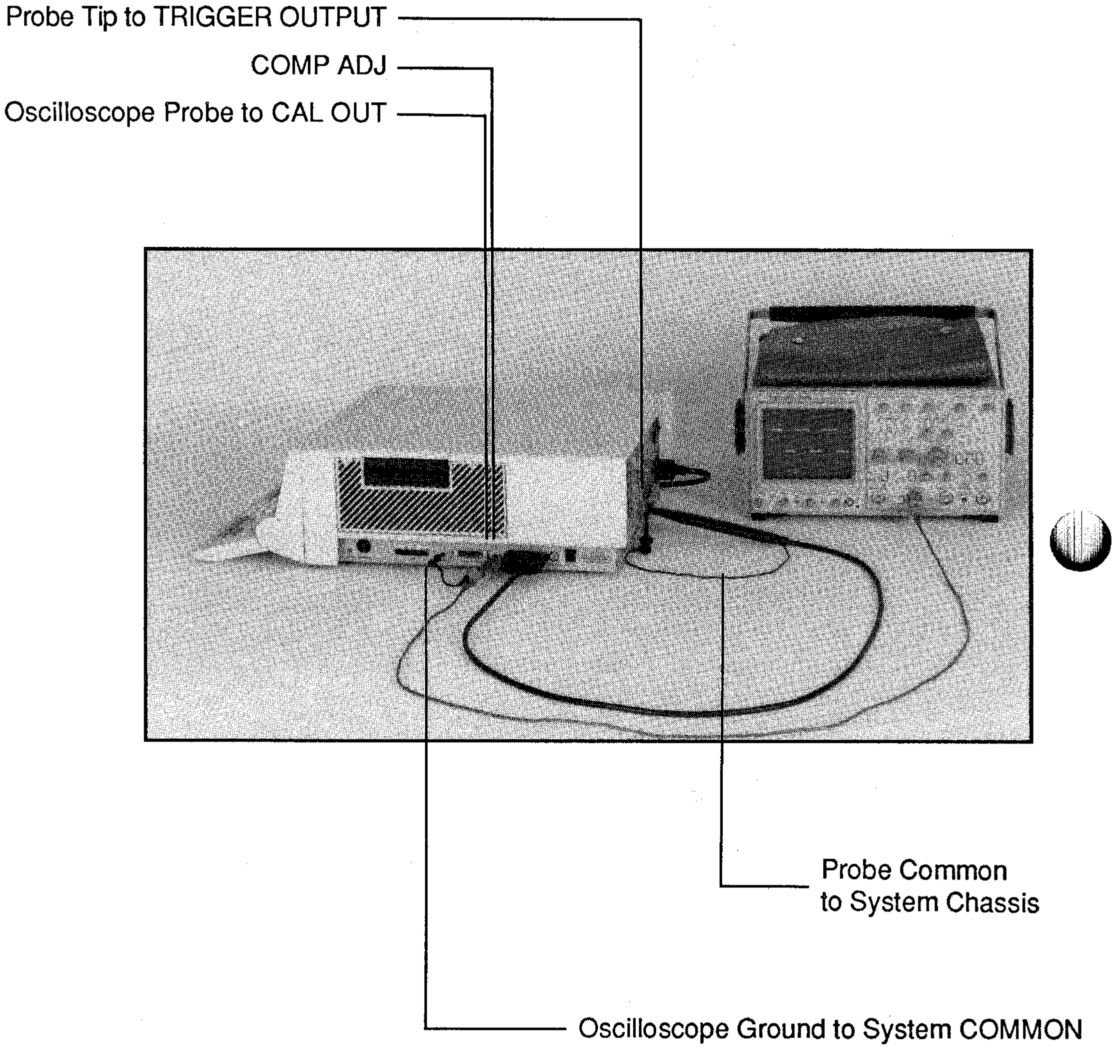

Probe Tip to TRIGGER OUTPUT
COMP ADJ
Oscilloscope Probe to CAL OUT
Probe Common to System Chassis
Oscilloscope Ground to System COMMON

Figure 4-1: Oscilloscope Use in Probe Compensation

3. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

4. Press the CAL softkey.

5. Move the cursor to the next field and press the PROBE softkey.

6. Move the cursor to the next field and press the COMP softkey. The display should read:

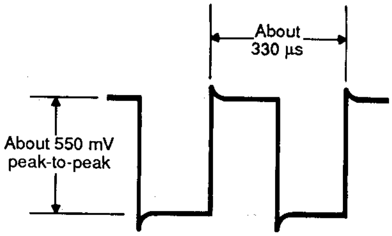
```
MAIN: CAL PROBE COMP
```

7. Press the ENTER key. The display should now read:

```
MAIN: CAL PROBE COMP
CONNECT PROBE TO TRIGGER OUTPUT
ADJUST COMP, PRESS STOP WHEN DONE
```

8. Insert the probe tip into the central (innermost) conductor of the TRIGGER OUTPUT at the rear of the 9100A/9105A. Leave the probe in that position.

9. Adjust the oscilloscope's horizontal and vertical settings, until an approximately square wave is displayed. Then use a trimmer tool on COMP ADJ to obtain the best square wave possible. The trimmer tool bears Fluke Part number 800540 or JFD 5284. The figure below shows the approximate waveform that should appear on the oscilloscope display.



10. Press the STOP key on the operator's keypad when you have finished the adjustment of Step 9. The display should read:

```
MAIN: CALIBRATION COMPLETE
```

Probe to External (Clock Module) 4.4.2.

This calibration automatically calibrates the probe's internal data delay to the external clock delay. The clock signal input is through the clock module, which must be connected to the system. To perform the calibration:

1. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

2. Press the CAL softkey.

3. Move the cursor to the next field and press the PROBE softkey.

4. Move the cursor to the next field and press the TO EXT softkey. The display should read:

   ```
   MAIN: CAL PROBE TO EXT
   ```

5. Press the ENTER key. The display should read:

   ```
   MAIN: CAL PROBE TO EXT
    CONNECT EXTERNAL CLOCK TO PROBE TIP AND
    COMMON LINES TOGETHER.  PRESS BUTTON
   ```

6. Connect the probe to the clock module CLOCK line.

7. Connect the probe's common clip to the clock module's COMMON line.

8. Press the probe ready button. The display should read:

   ```
   MAIN: CALIBRATION COMPLETE
   ```

Probe to Pod 4.4.3.

This calibration automatically calibrates the probe's internal data delay to the pod's PodSync line, which the system sometimes uses as a clock signal. To perform the calibration:

1. Connect a UUT to the pod.

2. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

3. Press the CAL softkey.

4. Move the cursor to the next field and press the PROBE softkey.

5. Move the cursor to the next field and press the TO POD softkey. The display will read something like that shown below. The last two fields are pod dependent and softkey selectable.

```
MAIN: CAL PROBE TO POD ADDR RISING
```

6. Move the cursor to he next field and press the desired softkey. For example, if you intend to use the 9100A/9105A in SYNC PROBE TO POD ADDR mode, you should press the ADDR softkey. The display should then read:

```
MAIN: CAL PROBE TO POD ADDR RISING
```

7. Move the cursor to the next field and press the desired softkey. For example, to calibrate on the rising edge (the default), press the RISING softkey. The default edge to use is always the first softkey choice.

8. Connect the probe common clip to the UUT common.

9. Press the ENTER key. A pod-dependent message is displayed like the following example:

```
MAIN: CAL PROBE TO POD ADDR RISING
 CONNECT PROBE TO ~S1
 PRESS PROBE BUTTON WHEN READY
```

10. Probe the specified pod line as directed. You may need to refer to the UUT schematic to find a convenient point at which you can do so.

11. Press the probe button when the probe tip is touching the point being probed. When calibration is complete, the display should read:

```
MAIN: CALIBRATION COMPLETE
```

12. Repeat steps 6 through 11 for each SYNC mode in which the 9100A/9105A is to be operated.

I/O Module to External 4.4.4.

This calibration calculates the proper setting for the I/O module's internal clock delay for use whenever the SYNC I/O MOD TO EXT command is entered. This calibration requires the use of the calibration module supplied with the I/O module. To perform the calibration:

1. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

2. Press the CAL softkey.

3. Move the cursor to the next field and press the I/O MOD softkey.

4. Move the cursor to the next field and press the EXT softkey. The display should read:

```
MAIN: CAL I/O MOD TO EXT
```

5. Press the ENTER key. The display should read:

```
MAIN: CAL I/O MOD TO EXT
 INSTALL CAL HEADER IN DESIRED I/O MODULE
 PRESS BUTTON WHEN READY
```

6. Fit the calibration module over the I/O module to be tested.

7. Press the ready button on the calibration module. When the calibration is complete, the BUSY light should go off and the display should read:

```
MAIN: CALIBRATION COMPLETE
```

I/O Module to Pod 4.4.5.

This calibration calculates the proper settings for the I/O module's internal clock delay for use with the SYNC I/O MOD TO EXT and SYNC I/O MOD TO POD command. When either command is entered, the appropriate delay is selected. This calibration requires the use of the calibration module. To perform the calibration:

1. Connect a UUT to the pod.

2. Press the MAIN MENU key and use the left arrow key to move the cursor to the left-most field.

3. Press the CAL softkey.

4. Move the cursor to the next field and press the I/O MOD softkey.

5. Move the cursor to the next field and press the POD softkey.

6. Move the cursor to the next field and press the desired softkey. For example, if you intend to use the 9100A/9105A in SYNC I/O MOD TO POD ADDR mode, you should press the ADDR softkey. The display should then read:

```
MAIN: CAL I/O MOD TO POD ADDR RISING
```

7. Move the cursor to the next field and press the desired softkey. For example, to calibrate on the rising edge (the default), press the RISING softkey. The default edge to use is always the first softkey choice.

8. Press the ENTER key. The display should read:

```
MAIN: CAL I/O MOD TO POD ADDR RISING
 INSTALL CAL HEADER IN DESIRED I/O MODULE
 PRESS BUTTON WHEN READY
```

9. Plug the calibration module into the I/O module, and press the calibration module's ready button. Make sure the calibration lead on the calibration module is unconnected when pressing the ready button.

10. After a few seconds, a pod-dependent message will be displayed. For example, the display may read:

```
COMPLETED EXT CAL PRIOR TO CAL POD
 NOW CONNECT CAL LEAD TO ~S1
 PRESS BUTTON WHEN READY
```

11. Refer to a schematic of your UUT and locate the specified signal: in this case, ~S1. At a suitable point on the UUT, attach the calibration lead to this signal.

12. Press the calibration module's ready button. After several seconds, the display should read:

```
MAIN: CALIBRATION COMPLETE
```

13. Repeat steps 6 through 12 for each SYNC mode in which the 9100A/9105A is to be operated.

AN ALTERNATIVE TO CALIBRATION 4.5.

You do not have to calibrate the system at every power-up. A more convenient procedure is to restore calibration data from the user disk after the self-tests have been performed and the system configured. If you do not calibrate the system at power-up, you should at least restore saved calibration data.

Each calibration generates data that can be saved on the user disk using the SETUP MENU key. Once the system is calibrated for a given pod, probe, clock module, and I/O module, the data is good until one or more of those devices is changed.

If you test or troubleshoot one type of UUT and then change to another, the calibration data stored on user disk for the new UUT will be invalid if a new pod is connected. In this case, the system should be recalibrated, after which the new data can be stored on the user disk.

Invalid Calibration Data 4.5.1.

The calibration data you restore from user disk may be invalid if:

- After the data was saved, your system's pod, probe, I/O modules, or clock module were changed. This includes swapping I/O modules: for example, removing an I/O module from connector 1 and attaching it to connector 2.
- Your system has never been calibrated.
- Your system was last calibrated more than one month ago.
- When testing or troubleshooting, you changed pods or I/O modules when changing from one type of UUT to another.

Saving Calibration Data 4.5.2.

Save calibration data on disk as follows:

1. Press the SETUP MENU key and use the left arrow key to move the cursor to the left-most field.

2. Press the SAVE softkey.

3. Move the cursor to the next field and press the CALDATA softkey.

4. Move the cursor to the next field and press one of the following softkeys:

   USERDISK: Used if the user disk does not contain a UUT directory. The display should then read:

   ```
   SAVE CALDATA IN USERDISK
   ```

   UUT FILE: Used if the user disk contains a directory for the UUT.

5. If you pressed the UUT FILE softkey in step 4, type the UUT directory name: for example, type "DEMO". The display should read:

   ```
   SAVE CALDATA IN UUT FILE DEMO
   ```

6. Press ENTER to save calibration data.

Restoring Calibration Data 4.5.3.

To restore calibration data from a user disk:

1. Press the SETUP MENU key and use the left arrow key to move the cursor to the left-most field.

2. Press the RESTORE softkey.

3. Move the cursor to the next field and press the CALDATA softkey.

4. Move the cursor to the next field and press one of the following softkeys:

   USERDISK: If the calibration data was saved in the USERDISK directory. The resulting display should read:

   ```
   RESTORE CALDATA FROM USERDISK
   ```

   UUT FILE: If the calibration data was saved in a UUT directory.

5. If you pressed the UUT FILE softkey in step 4, type the UUT directory name: for example, type "DEMO". The display should read:

   ```
   RESTORE CALDATA FROM UUT FILE DEMO
   ```

6. Press ENTER to restore the previously saved calibration data.

SUMMARY 4.6.

Self-tests: Should be performed for each device (pod, probe, or I/O module) when the device is attached to the mainframe.

Configurations: Can be saved on and restored from user disk for each UUT. They specify SETUP MENU command settings and provide information for other commands.

Calibration: Generates data which can be stored on user disk and restored at power-up or reset. This data is sometimes invalid. Calibration must be performed when:

- The system is first installed and every 30 days thereafter.
- The pod, probe, I/O module, or clock module is changed.
- The system is powered up or reset.

Before you test or troubleshoot a UUT, you should ensure that self-tests, calibration, configuration, and disk drive verification have all been performed appropriately. Some guidelines are as follows:

After each power-up or reset before you test or troubleshoot a UUT:

- Calibrate and configure the system for the type of UUT you are about to test or troubleshoot.
- Ensure that self-tests have been performed on the pod, probe, and I/O modules.
- Verify that the system mode is appropriate and that your user disk is in the correct drive.

If you reset or change any part of the system:

- The self-tests will have to be repeated and the system recalibrated. You can save new calibration data on the user disk, writing over the old data.
- Test and troubleshooting programs and the system configuration remain valid.

If you change UUTs from one type to another:

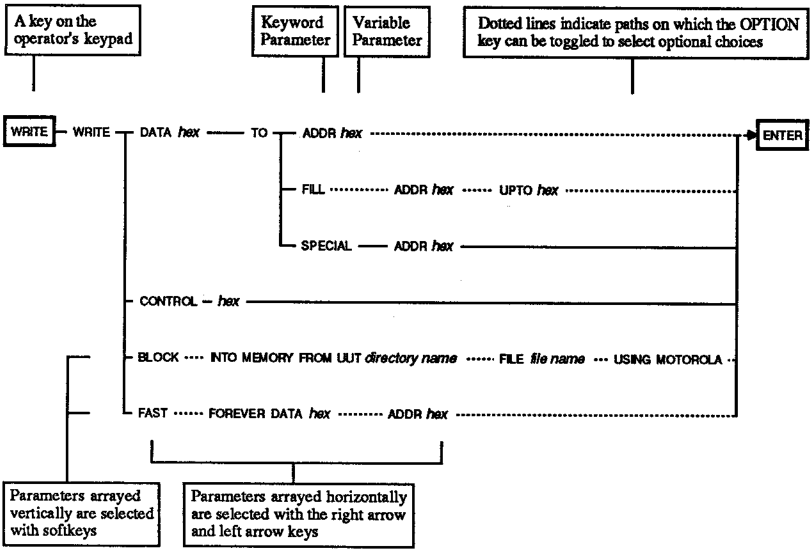
- You may have to change pods or I/O modules. If so, repeat the self-tests and calibrations with the new pod.
- System settings are probably different for the new UUT. Restore settings from the new UUT directory to reconfigure the system.

Section 5
Keypad Reference

This section is a detailed, alphabetical reference to all keys on the operator's keypad. You should already have read Section 3, "Typical 9100A/9105A Operations," to learn the basic use of the keypad.

Figure 5-1 shows how to read a syntax diagram. A syntax diagram is shown for each key in this section, and all the diagrams appear together in Appendix A, "Keypad-Syntax Quick Guide."

The text explanations for each command are, in general, indented to make them resemble the syntax diagrams. It would be useful to refer to the relevant diagram while reading an explanation. In this manual, "using" a key means to execute the commands associated with that key.


A key on the operator's keypad
Keyword Parameter
Variable Parameter
Dotted lines indicate paths on which the OPTION key can be toggled to select optional choices
WRITE
WRITE
DATA hex
TO
ADDR hex
ENTER
FILL
ADDR hex
UPTO hex
SPECIAL
ADDR hex
CONTROL
hex
BLOCK
INTO MEMORY FROM UUT directory name
FILE file name
USING MOTOROLA
FAST
FOREVER DATA hex
ADDR hex
Parameters arrayed vertically are selected with softkeys
Parameters arrayed horizontally are selected with the right arrow and left arrow keys

Read diagrams from left to right. Horizontal parameters are selected with the right arrow and left arrow keys. Vertical parameters are selected with softkeys.

Variables Parameters

.... A decimal number that you type on the keypad. The softkeys INC (increment) and DEC (decrement) appear when this field is selected.

hex .... A hexadecimal number that you type on the keypad. The softkeys INC (increment) and DEC (decrement) appear when this field is selected.

file name .... An alphanumeric name of a program or text file (e.g., "Test5").

component name .... An alphanumeric name of a component (e.g., "U55").

directory name .... An alphanumeric name of a file directory for a specific type of UUT (e.g., "DEMO").

pod name .... An alphanumeric name of a pod (e.g., "80286").

Figure 5-1: Interpreting Keypad Syntax Diagrams

ALPHA 5.1.

Changes the keypad from alphanumeric mode to command mode and turns the red ALPHA light off.

ALPHA

The keypad keys do different things in each mode:

- Command mode: The white keys produce hex digits 0 through 9 and A through F. The grey keys produce the functions indicated on them in black.
- Alphanumeric mode: All white keys and most grey keys produce ASCII characters. Use the white keys to type characters 0 through 9 and A through F. Use the grey keys to type characters G through Z, a space, an underscore, or a period. The characters are indicated in red on the grey keys.

The ALPHA key is automatically activated when the cursor is over an alphanumeric field. It cannot be activated at other times. When the ALPHA light is on, you can press the HELP key to display a list of possible entries. The up arrow and down arrow keys allow you to scroll through the list.

Section 3, "Typical 9100A/9105A Operations," contains an example demonstrating the use of this key.

BUS 5.2.

Tests the UUT power supply and system bus. You may need to use the OPTION key before pressing ENTER to set the address option to MEMORY.

BUS ·············· TEST BUS AT ADDR ······ *hex* ··························→ ENTER

TEST BUS AT ADDR hex

The bus test checks that each address and data line can individually be driven high and low. It also verifies that no address or data lines are tied together or stuck at a fixed level. This test should be the first one performed when troubleshooting a UUT.

The address specified is needed to test the data lines. The address should be in memory—i.e., in RAM, not in ROM nor I/O space. Other locations (if any are allowed) should only be specified if they physically exist and are not written to by other components.

UUT faults discovered by this test always produce fault messages, even if fault reporting functions are turned off using the SETUP MENU key. The fault messages are described in Appendix F of this manual.

CLEAR/NO 5.3.

Serves as a NO response to a displayed question. Also used when typing characters into a field to delete the last character typed.

CLEAR NO

You cannot type characters into softkey-selectable fields, and you cannot use this key with such fields.

Most fields assume default values on power-up or reset. When you first start typing characters into such fields, the previous contents vanish, and are replaced by what you type. If you delete all the characters you typed, the former contents will reappear.

Some alphanumeric fields have blank defaults. You type characters into them and use the CLEAR key as you do for other fields.

CONT 5.4.

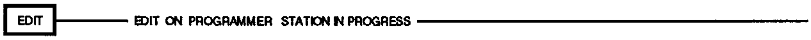
Causes a program or a built-in test to continue execution after stopping. Used after the red STOPPED light comes on.

CONT

The system halts execution of programs or built-in tests if the STOP key is pressed. Execution may also halt if the operator has to input data, or if a UUT fault is encountered. If the STOP key is pressed or a UUT fault is encountered, the STOP light comes on. Press CONT to resume execution.

EDIT 5.5.

Is available on the 9100A only. Transfers control of the system to the programmer station (monitor and keyboard).



EDIT ON PROGRAMMER STATION IN PROGRESS

While in EDIT mode, the operator's display and keypad are deactivated. To escape this mode, simultaneously press Shift and Quit on the programmer's keyboard.

ENTER/YES 5.6.

Serves as a YES response to questions on the operator's display or causes a command to be performed as displayed.

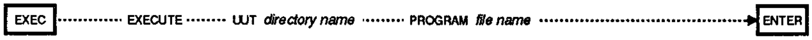
ENTER YES

Some commands allow address option selection using the OPTION key before pressing ENTER. You should be aware of existing options, which are displayed only if options are toggled on by pressing the OPTION key.

After you change fields, the fields retain their new values until changed again. Switching off the system or pressing RESET sets all fields to power-up values.

EXEC 5.7.

Executes a program from a specified directory.



If other parameters are needed, they are requested after you press ENTER.

EXECUTE UUT <directory name> PROGRAM <file name>

Specifies the UUT directory and a program within the directory. Pressing the ENTER key causes the program to be executed.

The UUT directory contains all programs and text files associated with the UUT. Section 3 contains examples showing how pre-programmed tests are run using this key.

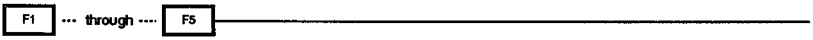
When you are prompted for the UUT directory name or the program file name, pressing the HELP key will display a list of available UUT directory names or program file names, respectively.

NOTE

If you are prompted for any numeric parameters after pressing the ENTER key, they must be entered as hexadecimal numbers.

F1 ... through ... F5 (softkeys) 5.8.

Pressing a softkey selects the function specified by the softkey label on the display directly above the softkey. The field the cursor is at changes to the selected function. A softkey is only active if a label is displayed above it.

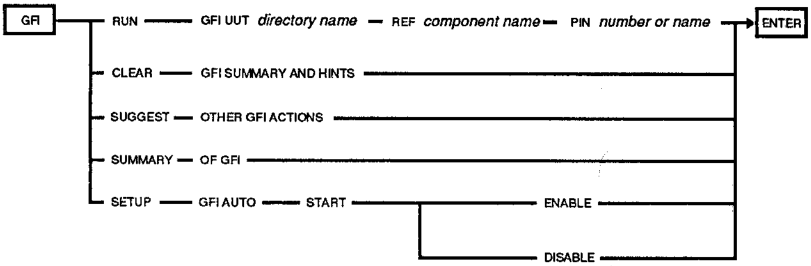


A set of up to five softkey labels can be displayed at one time. If a field has more than five softkey options, extra sets of labels are hidden from view. In such cases, the MORE SOFTKEYS light comes on, reminding you to scroll hidden sets of labels into view by pressing the SOFT KEYS key as often as necessary. After the last set is in view, repeatedly pressing this key then scrolls the labels back towards the first set.

If there are only two sets of labels, pressing this key would toggle alternate sets into view.

GFI 5.9.

Executes the GFI algorithm to troubleshoot a particular UUT and stores a summary of the results. GFI can begin execution at any UUT location specified, or if the UUT previously failed a functional test, GFI can start at a location suggested by the test.



Before GFI can troubleshoot any UUT, information about the UUT must be stored in a database within the directory containing all files associated with that UUT. The database is set up in EDIT mode using the programmer's station. Section 5 in the *Programmer's Manual* contains details on the GFI algorithm and on how to create or enter information into the GFI database.

When you are prompted for the UUT directory name, pressing the HELP key will display a list of available UUT directory names.

RUN GFI FOR UUT <directory name> REF <component name> PIN <pin name or pin number>

Enter the UUT name, component name, and the pin to be tested. When GFI has a probing suggestion, the component name and pin number are displayed as default values. Such suggestions are generated by previous GFI activity or by earlier functional tests.

When you press ENTER, you will be asked to clip or probe the component. When clipping it:

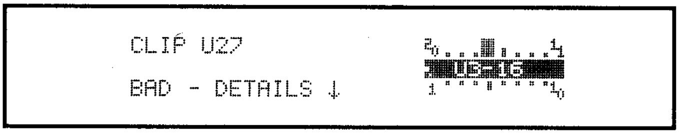
- The clip module must be aligned so that its pin 1 (marked by the grey wire) is on component pin 1.
- You can use a clip larger than the component.

After positioning the clip module or probe, press the device's ready button; when the button is pressed, the pin on the specified component will be tested and the results displayed. In addition, if the testing device is an I/O module (clip module), *all* pins on the specified component will be tested and the results displayed.

Figure 5-2 shows typical displays of GFI results. In the top left corner is a suggestion or an accusation: in Example 1, a suggestion, "CLIP U27". The right hand side of the display represents the IC just tested. The arrows indicate bad pins. The bottom left corner displays a status message for each pin.

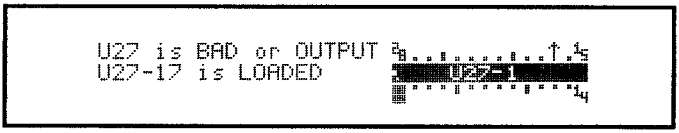
You can use the left and right arrow keys to move the cursor to each of the IC pins. As the cursor moves, the status message displayed in the lower left corner changes, reflecting the state of the current pin. If the pin is bad, the message and the MORE INFORMATION light indicate that you should use the up and down arrow keys to scroll through a long message.

The recommendation "CLIP U27" in Example 1 indicates that GFI has decided to test U27 next. To test U27, clip it and then press the I/O module ready button. To override the suggestion,



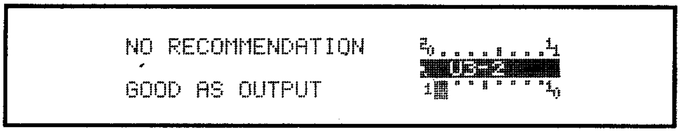
```
CLIP U27
BAD - DETAILS ↓
```

Example 1: GFI Makes Probing Recommendation



```
U27 is BAD or OUTPUT
U27-17 is LOADED
```

Example 2: GFI Accuses Probed Component



```
NO RECOMMENDATION
GOOD AS OUTPUT
```

Example 3: GFI Makes No Recommendation

Figure 5-2: Example Results of the GFI RUN Command

enter a RUN GFI command with the component name and pin number you desire.

CLEAR GFI SUMMARY AND HINTS

Erases the hint list, all information loaded from the GFI database, and the locally stored GFI summary. The RESET command will not do this.

This command should be used each time you begin troubleshooting a UUT to avoid potential confusion between an old UUT and the new one. CLEAR GFI can also be used if you have to restart GFI after a mistake.

SUGGEST OTHER GFI ACTIONS

Displays a comprehensive list of GFI suggestions. Example 1 in Figure 5-3 shows a sample suggestion list. The list, generated from previous GFI activity, includes accusations, probing recommendations, and unused hints from any previous functional test that the UUT failed. The command displays alternatives to the current recommendation. The current recommendation is also the default value of the GFI RUN command.

The MORE INFORMATION light is on in Example 1 to indicate that the up and down arrow keys can be used to scroll through the list.

SUMMARY OF GFI

Displays a summary of GFI activity; the summary is stored in local memory, and not on disk. Example 2 in Figure 5-3 shows a typical GFI summary. It contains one entry for each component that had at least one pin tested by GFI. The summary lists the name of each component along with the number of bad inputs, bad outputs, and untested pins on each.

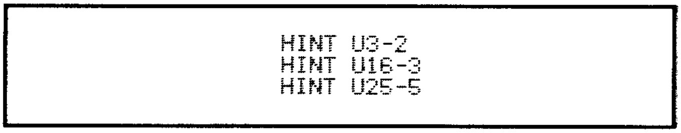
SETUP GFI AUTO START . . .

ENABLE

Enables an automatic transition from a TL/1 functional test to GFI. If GFI autostart is enabled, GFI automatically starts after a TL/1 program that generates GFI hints has finished running.

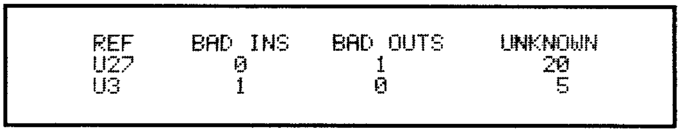
DISABLE

Disables the automatic startup of GFI.



```
HINT U3-2
HINT U16-3
HINT U25-5
```

Example 1: GFI Suggestion List



```
REF     BAD INS    BAD OUTS     UNKNOWN
U27        0          1            20
U3         1          0             5
```

Example 2: GFI Summary

Figure 5-3: GFI Summary and Suggestion List Examples

HELP 5.10.

Turns the HELP window on and off. The HELP window covers the display and provides information about the fault condition displayed.

If the previous display was a fault message, pressing the HELP key displays an explanation of the message. Use the up arrow and down arrow keys to scroll through the message.

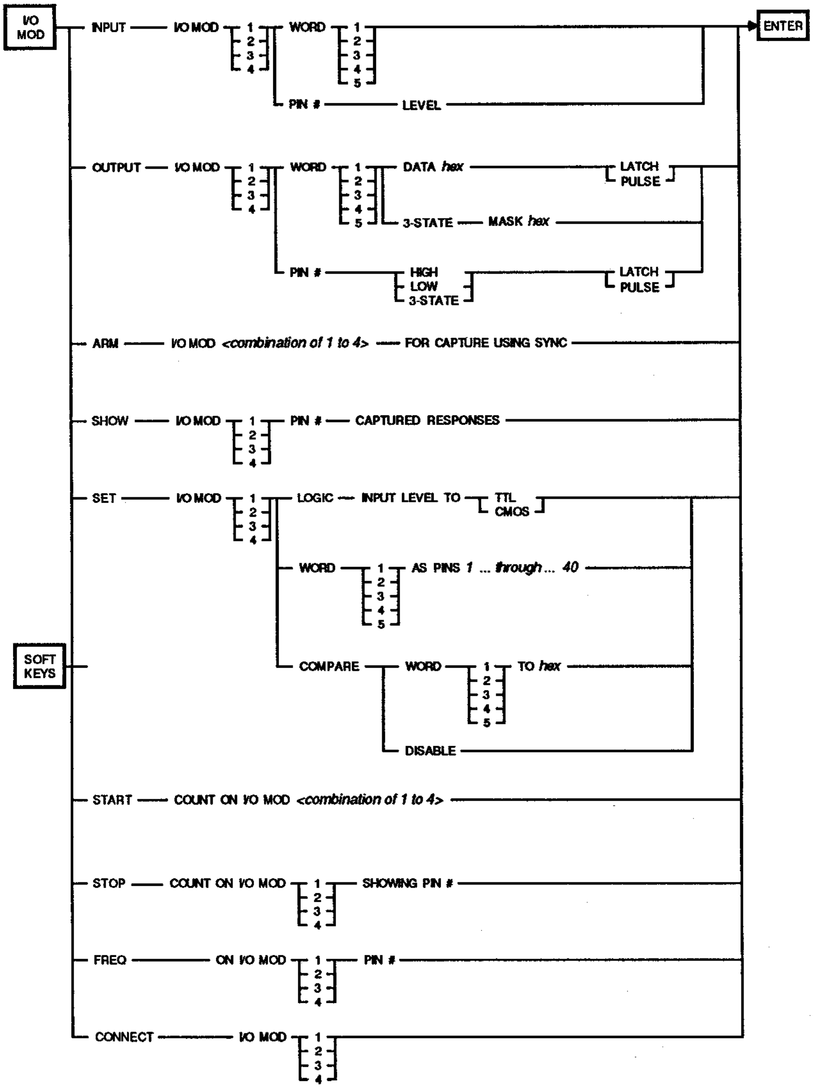
HELP

The HELP key provides additional ALPHA field selection features. The key displays possible ALPHA type selections and allows automatic entry of each ALPHA selection. When the cursor is in a UUT directory, PROGRAM, POD NAME, TEXTFILE, or FILE entry field, pressing the HELP key displays the possible selections on the lower two lines of the display. An inverse cursor is placed on the first selection and the entry on the first line is changed to indicate that selection. The arrow keys move the inverse cursor to each of the selections, scrolling the last two lines if more are available, and automatically change the entry in the first line field. To indicate that the present entry is correct, press the ENTER key. To exit from this selection mode and return the previous entry, press the CLEAR or STOP key. After the entry is selected using the ENTER key, press the ENTER key again to execute the desired command. The following keys have ALPHA selections: EXEC, GFI, READ (BLOCK softkey), WRITE (BLOCK softkey), MAIN MENU (PRINT, REMOVE, and EDISK softkeys), and SETUP MENU (RESTORE and SAVE softkeys).

I/O MOD 5.11.

Causes outputs to and inputs from the UUT. Data is input or output through the I/O modules when a command is entered or at times set by the SYNC command.

I/O MOD Key


I/O MOD
INPUT
I/O MOD
1
2
3
4
WORD
1
2
3
4
5
PIN #
LEVEL
ENTER
OUTPUT
I/O MOD
1
2
3
4
WORD
1
2
3
4
5
DATA hex
LATCH
PULSE
3-STATE
MASK hex
PIN #
HIGH
LOW
3-STATE
LATCH
PULSE
ARM
I/O MOD <combination of 1 to 4>
FOR CAPTURE USING SYNC
SHOW
I/O MOD
1
2
3
4
PIN #
CAPTURED RESPONSES
SET
I/O MOD
1
2
3
4
LOGIC
INPUT LEVEL TO
TTL
CMOS
WORD
1
2
3
4
5
AS PINS 1 ... through ... 40
SOFT KEYS
COMPARE
WORD
1
2
3
4
5
TO hex
DISABLE
START
COUNT ON I/O MOD <combination of 1 to 4>
STOP
COUNT ON I/O MOD
1
2
3
4
SHOWING PIN #
FREQ
ON I/O MOD
1
2
3
4
PIN #
CONNECT
I/O MOD
1
2
3
4

I/O Module Capabilities 5.11.1.

An I/O module can perform three types of operations: input, output, and comparisons. Most operations can be done through individually specified pins or through a user-defined word (ordered set of pins).

- Input (from the UUT): An I/O module can gather the following types of responses from the UUT:
  - Logic levels (INPUT WORD or INPUT PIN). Displayed as high (1), low (0), or invalid (X).
  - Asynchronous level history (INPUT PIN or ARM followed by SHOW). Displays all levels observed at the specified pin input since the last command affecting that module.
  - Clocked level history (ARM followed by SHOW). Displays all levels observed during clock pulses at the specified pin since the last execution of a command affecting the I/O module.
  - CRC signatures (ARM followed by SHOW).
  - Input frequency (FREQ). Frequency of rising transitions at the specified pin. Rising transitions are defined below.
  - Transition count (START COUNT followed by STOP COUNT). Rising transitions are counted at the specified pin. Rising transitions are low-to-high and invalid-to-high transitions.
- Output (to the UUT): Outputs can be high, low, or high impedance. They can be latched or pulsed.
- Comparisons: Response data words can be asynchronously compared to a previously stored word. Matches are indicated on the display and also by a high level on the DCE pin at the side of the I/O module.

Examples of I/O module control are found in the SYNC key section and in Section 6, "Test Techniques and Examples."

I/O Module Synchronization 5.11.2.

Figure 2-9 showed how one I/O module interfaces with its external sync lines, the system, the internal PodSync line, and the UUT.

Information can flow between the UUT and the system one bit at a time (specific pins) or in groups of bits (words). Each module can manipulate up to five words. A word is limited to 40 bits, the number of pins on each module.

The I/O module can gather data at the following times:

- At entry of any of these I/O MOD commands: INPUT, FREQ, or STOP COUNT after a previous START COUNT.
- After ARM I/O MOD is entered and until SHOW I/O MOD is entered as synchronized by the SYNC command.

The SYNC command synchronizes I/O module data gathering to one of the following clocked modes:

- **System freerun clock:** This mode is only of use under TL/1 program control.
- **Pod's PodSync line.** Response gathering occurs at the trailing (rising) edge of PodSync, an internal signal generated by selected pod cycles.
- **I/O module external sync lines .** The external START, STOP, ENABLE, and CLOCK input control lines (from the UUT) control the I/O module data gathering.

Start, Stop, and Enable conditions are defined using the SYNC I/O MOD TO EXT command. The COMMON line is always connected to UUT common. The CLOCK line, which is modified by Start and an enabling condition, is connected to the desired UUT clock.

- **Internal:** This mode should only be used under control of a TL/1 program, which can generate an internal clock signal for synchronization.

Timing diagrams and more detailed explanations of synchronization are provided in the SYNC section. START, STOP, ENABLE, CLOCK, and COMMON are also available as pins at the side of the I/O module. The pins can be used to daisy-chain up to four I/O modules with the last one in the chain being connected to the UUT. DCE (data compare equal) is an output and should not be daisy-chained.

I/O Module Word Definition 5.11.3.

A word is a set of I/O module pins, ordered so as to map in a known way to clip module pins, and hence to the pins of the IC being clipped over. Appendix B lists this mapping for all I/O module pins and clip modules.

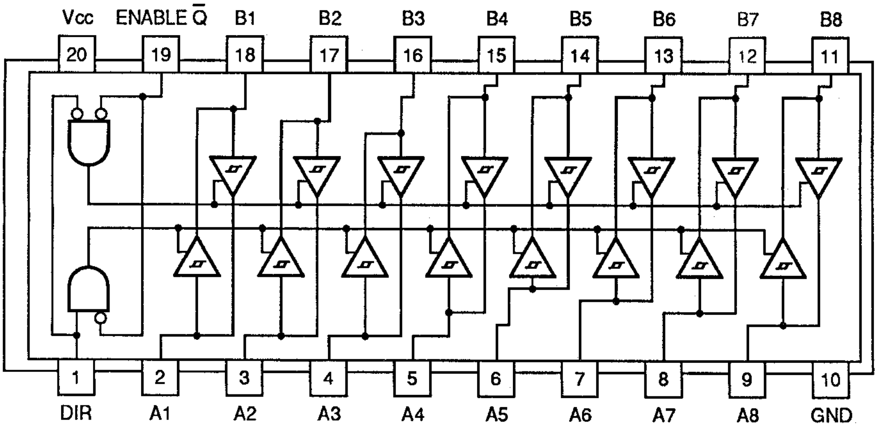
The last entry of an I/O module word definition should usually be "0" inidicating that any following pins are not to be included. For example, if you enter the command:

```
SET I/O MOD 2 WORD 5 AS PINS 1 2 3 0
```

the I/O module word will include only I/O module pins 1, 2, and 3. All other pins that were displayed are set to zero and ignored.

Figure 5-4 shows a 74LS245 octal bus transceiver. To define pins A1 through A8 as "word 5" on I/O module 4:

1. Verify the type of IC: 20-pin DIP package.



74LS245 Octal Bus Transceivers

IC Pin Function	IC Pin Number	Clip Module Pin Number[1]	I/O Module Pin Number[2]
A8 (MSB)	9	9	29
A7	8	8	28
A6	7	7	27
A5	6	6	26
A4	5	5	25
A3	4	4	24
A2	3	3	23
A1 (LSB)	2	2	22

[1] Assume that the clip module exactly fits the IC.

[2] See Step 4.

Typical I/O module WORD pin mapping

Figure 5-4: I/O Module Word Definition Example

2. Write down the pin numbers that are to form the word, starting with the most-significant bit (A8 by convention):

   ```
   74LS245
   Function:  A8  A7  A6  A5  A4  A3  A2  A1
   IC Pin
   Number:     9   8   7   6   5   4   3   2
   ```

3. If the IC has 24 pins or less, decide which side of the I/O module ("A" or "B") to use. If the IC is 28 pins or more, the clip module occupies both sides. In this example, the "B" side is selected.

4. Look up the 20-pin "B" side clip module mapping in Appendix B, and write down the I/O module pins corresponding to the clip module pin numbers, which are normally the same as the IC pin numbers in step 2:

   ```
   Clip
   Mod Pin:     9   8   7   6   5   4   3   2
   I/O
   Mod Pin:    29  28  27  26  25  24  23  22
   ```

5. Enter the command:

   ```
   SET I/O MOD 4 WORD 5 AS PINS 29 28 27 ...
   ... 26 25 24 23 22 0
   ```

Word 5 does not always have to be used with a 20-pin clip module. You should however be aware that the I/O module pins will not map the same way to another clip module, or to the same clip module on the "A" side.

How to Output Data to a Word 5.11.4.

If we assume that Word 5 on I/O module 4 was defined as shown in the previous section, and we assume that pins A1 through A8 of the 74LS245 are already set to the high impedance state by the UUT, then we can output data to these pins as follows:

- **To latch data word AB** (1010 1011) to word 5, use the command OUTPUT I/O MOD 4 WORD 5 DATA AB LATCH. The data will be fixed at the following levels:

```
74LS245 Pin:             9  8  7  6    5  4  3  2
Word 5, I/O Mod Pin:    29 28 27 26   25 24 23 22
Data at Pin:             1  0  1  0    1  0  1  1
```

- **To force pins 4 and 5 to high-impedance** (X) level, use the command OUTPUT I/O MOD 4 WORD 5 3-STATE MASK 0C. Note that bits 4 and 5 are set in the hex word 0C (0000 1100).

```
74LS245 Pin:             9  8  7  6    5  4  3  2
Word 5, I/O Mod Pin:    29 28 27 26   25 24 23 22
Data at Pin:
(Before command):        1  0  1  0    1  0  1  1
(After command):         1  0  1  0    X  X  1  1
```

- **To pulse pin 6 high, pin 5 high, and pin 4 low,** enter the command OUTPUT I/O MOD 4 WORD 5 DATA BB PULSE. Hex data word BB is 1011 1011, The values for pin 6 (1), pin 5 (1) and pin 4 (0) are not the same as the existing data, which means that they will pulse as required.

```
74LS245 Pin:             9  8  7  6    5  4  3  2
Word 5, I/O Mod Pin:    29 28 27 26   25 24 23 22
Data at Pin:
(Before command):        1  0  1  0    X  X  1  1
(During command):        1  0  1  1    1  0  1  1
(After command):         1  0  1  0    X  X  1  1
```

These pulses are repeated each time the command is entered or repeated using the REPEAT key.

NOTE

In the previous example, if pins A1 through A8 are not at a high-impedance state before each OUTPUT command, the I/O module outputs would fight UUT outputs, possibly resulting in an I/O module over-current error message. Along with this message, all outputs of all I/O modules are set to high-impedance to protect both the UUT and the I/O modules from damage.

INPUT I/O MOD <n> ...

Specifies the I/O module from which to read level information. Information can be displayed for one pin or for a word defined by SET I/O MOD WORD. "n" is a softkey option: 1, 2, 3, or 4.

WORD <w>

Displays present levels as shown below for each pin in the specified word. The word should have been defined earlier using SET I/O MOD WORD. "w" identifies the word to be input. It is a softkey option: 1, 2, 3, 4, or 5.

```
I/O MOD 1 WORD 1 = A?3B
10101X1X00111011
```

Levels are shown as a hex number, and as a bit pattern. Invalid levels are shown by "X" in the bit pattern and "?" in the hex number.

INPUT I/O MOD <n> ...
(continued)

PIN # LEVEL

Displays the present level and asynchronous level history at the specified pin as shown below. The "X" (invalid) level is recorded in the LEVEL HISTORY only if the input stayed at that level for longer than 100 nsec. This condition does not apply to the present LEVEL of the first line.

```
I/O MOD 2 PIN 26 LEVEL = 1
LEVEL HISTORY          = 0X1
```

OUTPUT I/O MOD <n> ...

Specifies the I/O module to which data is to be written at command entry. "n" is a softkey option: 1, 2, 3, or 4. The command is not influenced by SYNC. Data can be written to one pin, or to a word defined by SET I/O MOD WORD.

At power-up or reset, all outputs are high-impedance. Outputs to specified pins or words can be subsequently latched or pulsed. To return a previously latched output to high-impedance, use OUTPUT I/O MOD WORD 3-STATE MASK.

WORD <w> ...

DATA hex LATCH/PULSE

Specify the data word to be output, as a 10-digit hex number. The data can be latched (fixed at the level until changed) or pulsed.

OUTPUT I/O MOD <n> ...

WORD <w> ...
(continued)

3-STATE MASK hex

Sets specified I/O module bits in the word to the high impedance state. The set bits in the output data word correspond to the specified lines. The STIM command section contains more details on mask calculation.

PIN # HIGH/LOW/3-STATE LATCH/PULSE

Drives the specified pin of the I/O module to the specified level (high or low) or sets it to high-impedance (3-state). Output is latched or pulsed.

ARM I/O MOD <combination of 1 to 4> FOR CAPTURE USING SYNC

Resets and arms (activates) all specified I/O modules for response gathering at each pin. For example, the command ARM I/O MOD 123 FOR CAPTURE USING SYNC would arm modules 1, 2, and 3 simultaneously. Other examples of "combination of 1 to 4" are: 1, 14, 134.

The following responses are gathered after the command is entered during times specified by the SYNC command:

- CRC signatures.
- Transition counts.
- Clocked level history
- Asynchronous level history.

ARM I/O MOD <combination of 1 to 4> FOR CAPTURE USING SYNC
(continued)

After entering this command, execute a manual or programmed stimulus and then enter SHOW I/O MOD to display resulting responses. A stimulus is a series of commands designed to generate known responses of the type described previously in "I/O Module Capabilities."

SHOW I/O MOD <n> PIN # CAPTURED RESPONSES

Displays the responses gathered after the ARM I/O MOD command was entered. If the response-gathering period had not ended, an error message is displayed. "n" is a softkey option: 1, 2, 3, or 4.

CAUTION

After entering ARM I/O MOD, a subsequent SHOW I/O MOD will display invalid responses if any of the following commands were entered after ARM and before SHOW:

I/O MOD command: ARM, SET, START, STOP, FREQ, or INPUT.

SYNC command: SYNC I/O MOD.

SET I/O MOD <n> ...

Identifies an I/O module and then changes or sets various features on it, as described below. "n" is a softkey option: 1, 2, 3, or 4.

LOGIC INPUT LEVEL TO TTL/CMOS

Selects the appropriate input voltage threshold level.

WORD <w> AS PINS <1 through 40>

Identifies a word and specifies up to 40 I/O module pin numbers that make up the word. The first pin number corresponds to the most-significant bit. "w" is a softkey option: 1, 2, 3, 4, or 5.

Pin numbers can be assigned arbitrarily, but may not be repeated within a word. An I/O module can manipulate five words. A 40-bit word is represented by a 10-digit hexadecimal number. Entering a pin number of 0 terminates the word. The default value of each word is: pins 40 through 1 in order (40 is the most-significant bit).

Examples at the beginning of this section show how to define a word and then output data to individual pins in it. Appendix B shows how I/O module pins are mapped to the clip module pins.

SET I/O MOD <n> ...
(continued)

COMPARE ...

WORD <w> TO hex

Asynchronously compares bit patterns in the specified response data word to the hex word. The response word should be defined earlier by SET I/O MOD WORD. If the two words are the same the fact is reported on the display, and the DCE (data compare enable) pin becomes active. Invalid (X) levels are considered to be low when making the comparison.

The ENABLE line on the I/O module can be used to qualify this feature. If qualification is not necessary, use SYNC I/O MOD EXT ENABLE ALWAYS.

Matches will be reported if detected, after every READ, WRITE, INPUT I/O MOD, OUTPUT I/O MOD, BUS, RAM, ROM, and STIM operation, until the feature is disabled by SET I/O MOD COMPARE DISABLE. After a match is detected, this command must be reentered for future use of the feature.

DISABLE

Disables the feature specified by the SET I/O MOD COMPARE WORD command.

START COUNT ON I/O MOD <combination of 1 to 4>

Starts counting rising transitions on all pins of the specified I/O modules. For example, the command START COUNT ON I/O MOD 124 would be effective for I/O modules 1, 2, and 4. Other examples of "combination of 1 to 4" are: 1, 14, 123.

CAUTION

After pressing the START softkey, do not press it again before pressing the STOP softkey. Doing so would (incorrectly) reset the count to zero.

START COUNT I/O MOD and STOP COUNT I/O MOD define an asynchronous counting operation that is not affected by the SYNC command.

STOP COUNT ON I/O MOD <n> SHOWING PIN

Stops counting rising transitions on the specified I/O module and displays the count for the specified pin. To see similar results for another pin, move the cursor to the PIN field, enter a new pin number, and press the ENTER key. This command is asynchronous and is not controlled by the SYNC command. "n" is a softkey option: 1, 2, 3, or 4.

STOP stops the transition count only on the specified I/O module. If you had started counts on multiple I/O modules, each must be stopped individually.

FREQ ON I/O MOD <n> PIN

Calculates rising-transition frequency at the input of the specified I/O module pin and reports the result as an input frequency. This command is asynchronous and is not influenced by the SYNC command. "n" is a softkey option: 1, 2, 3, or 4.

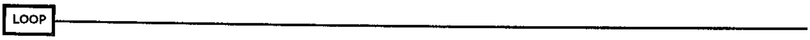
CONNECT I/O MOD <n>

The display shows how the I/O module external sync lines should be connected to component pins on the UUT. The information comes from the test program currently being run. The operator should connect the lines as indicated. An asterisk (*) on the display indicates that there has been "no change"; that is, the control line was connected to the same UUT component pin as of the last time the system prompted the operator to make a connection with that line. If all the connections are the same (none have changed since the last connect instruction), the connect command will not prompt the operator. "n" is a softkey option: 1, 2, 3, or 4.

The control lines are named START, STOP, ENABLE, CLOCK, and COMMON. The response-gathering parameters for these lines are set using the SYNC command.

LOOP 5.12.

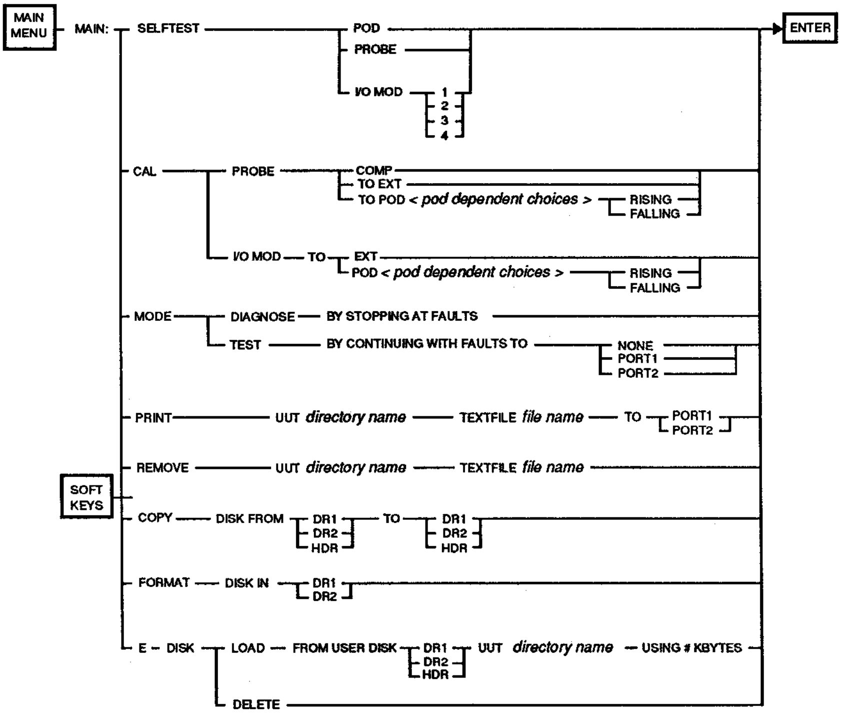
Repetitively performs the last pod test, probe, or I/O module operation specified. The repetition continues operation until STOP is pressed or a fault or error is encountered.



If a program or pod test halts after detecting a fault, the diagnostic sequence of instructions exercising the fault is executed repetitively. Pod-resident tests are invoked using the EXEC, POD, BUS, RAM, ROM, READ, WRITE, STIM, or RUN UUT keys. A test can have a fault handler (diagnostic routine) that is repetitively performed—instead of the entire test—when a fault occurs. Refer to the *Programmer's Manual* for information on TL/1 fault handlers.

MAIN MENU 5.13.

Performs miscellaneous functions, including self-tests, calibrations, fault reporting methods, disk formatting, and file manipulation.



MAIN: ...

SELFTEST ...

Executes built-in programs to test the pod, probe, and I/O modules. Failures generally mean that the device concerned must be serviced or repaired.

POD

Executes a comprehensive pod self-test program. If the pod fails, a message is displayed to explain the cause. If the message is not clear, look it up in the pod manual.

PROBE

Executes a pass/fail probe self-test program. Verifies that the probe is attached to and communicating with the system.

I/O MOD <n>

Executes a pass/fail self-test program for specified I/O modules. Verifies that the modules are attached to and communicating with the system.

The command MAIN: SELFTEST I/O MOD 23 results in self-tests being performed on modules 2 and 3.

CAL ...

Executes built-in calibration programs for the probe or I/O module. Generates calibration data which can be stored on and restored from disk, as explained in the section on calibration in Section 4. You can also refer to RESTORE CAL DATA and SAVE CAL DATA (SETUP MENU command).

MAIN: ...

CAL ...
(continued)

There are three basic types of calibrations performed on the 9100A/9105A:

- Compensation: matches impedance of a device to the impedance of its cable.
- Internal delay: Adjusts time taken by signals to travel a known path using a known reference. You do not have to perform this automatic calibration.
- Data delay against clock delay: Ensures that data arrives at the receiving hardware at the same time as the signal clocking the data.

PROBE ...

COMP

Calibrates probe compensation, matching the probe's impedance to that of the cable connecting it to the 9100A/9105A. Compensation requires the use of an oscilloscope. For details see Section 4.

TO EXT

Automatically calibrates the probe's internal data to the external clock delays using the external clock module, which is connected to the system.

TO POD <pod-dependent choices>

Automatically calibrates the probe's internal data delay to one or more pod-dependent PodSync lines.

MAIN ...

CAL ...

PROBE ...

TO POD <pod-dependent choices>
(continued)

A UUT must be connected to the pod to gain access to the PodSync signals. The user is prompted to probe the signals at the UUT. Calibrates to the selected edge (RISING/FALLING). The first softkey displayed is the default edge.

I/O MOD TO ...

EXT

Automatically calibrates the I/O module's internal data to external clock delays using the calibration module provided for each I/O module.

POD <pod-dependent choices>

Automatically calibrates the I/O module internal data delay to one or more pod dependent PodSync lines. A UUT must be connected to the pod to access the PodSync signals. The user is prompted to probe the required signals at the UUT. Calibrates to the selected edge (RISING/FALLING). The first softkey displayed is the default edge.

MODE ...

Specifies how faults are reported and whether or not the system stops executing commands when a fault is detected.

MAIN: ...

MODE ...

DIAGNOSE BY STOPPING AT FAULTS

Stops test execution when a fault is detected. This mode allows the operator to loop on the fault or enter commands to diagnose the fault. Pressing the CONT key resumes testing. This mode may be used when troubleshooting a UUT.

TEST BY CONTINUING WITH FAULTS TO DISPLAY ONLY/PORT1/PORT2

Outputs a fault message to the display only, or the display and serial port 1, or the display and serial port 2, and continues test execution. The message can be sent to a host computer or line printer connected to the specified port; in this mode, faults can be logged without an operator present.

PRINT UUT <directory name> TEXTFILE <file name> TO PORT1/PORT2

Outputs the specified text file at either serial port (1 or 2). You must specify both a UUT directory and a text file name to identify the file completely.

REMOVE UUT <directory name> TEXTFILE <file name>

Deletes a text file from the user disk. Specify a UUT directory and a file name to identify the file. The file is deleted from the disk in the drive currently specified by SETUP USERDISK (SETUP MENU command).

MAIN: ...

COPY DISK FROM DR1/DR2/HDR TO DR1/DR2/HDR

Copies the disk in the source drive to the disk in the destination. In a 9100A you can copy from the floppy disk drive (DR1) to the hard disk (HDR) and from the floppy disk drive (DR1) to itself. Copying from HDR to DR1 is possible but will usually result in a "media full" error because the hard disk has so much more storage capacity than the floppy disk. In a 9105A system, you can copy from one floppy disk drive (DR1 or DR2) to the other, or from either floppy disk drive to itself.

Copying from a drive to itself (DR1 to DR1, for example) is a sector-type copy and will completely write over any information currently on the destination disk.

Copying from one disk drive to another (DR1 to DR2, for example) is a merge/replace process. The examples below show the results of this process:

- File LATEST at the destination is written over by file LATEST from the source.
- File NEW on the source disk does not exist at the destination. File NEW is written to the destination.
- File OLD exists at the destination only. This file is not altered.

FORMAT DISK IN DR1/DR2

Erases the contents of the floppy disk in the specified drive and prepares the disk for storing files. You cannot format the hard disk.

CREATE/DELETE E-DISK 5.14.

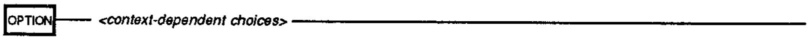
Creates or deletes the E-disk, a temporary RAM cache that TL/1 programs are loaded into to increase the speed of their operation. GFI information is also loaded onto the E-disk. You must specify the userdisk and the UUT directory containing the programs to be loaded onto the E-disk, along with the size of the E-disk in kilobytes. The LOAD operation creates the E-disk and loads all the TL/1 programs and the GFI database in the specified UUT directory into the E-disk. If a pod is specified by SETUP POD NAME (SETUP MENU command), all the TL/1 programs in the specified POD directory are also loaded. If no pod name has been set, these programs are not loaded.

The DELETE option deletes the E-disk, allowing the memory to be used for other operations. The E-disk is automatically deleted every time the EDIT key is pressed on the 9100A. The deletion process does not allow old versions of programs to be kept in the E-disk after being revised in the editor.

If not enough of memory is available for the E-disk, an "Out of Memory" error message is returned. Try the LOAD command again with a smaller amount of memory for the E-disk. If a message is returned informing you that the E-disk is too small to load in all the programs, reload a larger amount of memory for the E-disk until the error message stops appearing.

OPTION 5.15.

Specifies pod-dependent address options. This key turns the options display on and off. It is active only if the command to be executed has options. Options are not displayed automatically, but their presence is indicated in keypad syntax diagrams by dotted lines. The user should refer to Appendix C, "Pod-Related Information," for descriptions of options.



<context-dependent choices>

Address options are available only with certain fields in the following commands: BUS, RAM, READ, ROM, RUN UUT, STIM, and WRITE.

Options vary with the pod used. For example, a Z80 pod address space has to be specified as either memory or I/O. For users familiar with 9010 pods, the same addressing methods may be implemented using the READ SPECIAL and WRITE SPECIAL commands.

To select an option:

1. Display the command to be executed and specify all the fields appropriately for that command.

2. If the keypad is in alphanumeric mode, change the mode by pressing the ALPHA key.

3. Press the OPTION key. If available, options appear on the middle line of the display.

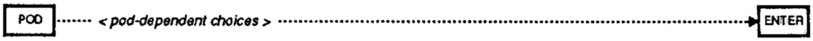
4. Press the down arrow key to place the cursor over the option field and display its softkey labels.

5. Press the appropriate softkey to change the field.

6. While the options remain displayed, press the ENTER key.

POD 5.16.

Selects miscellaneous pod-dependent functions, such as quick pod-resident tests when available.



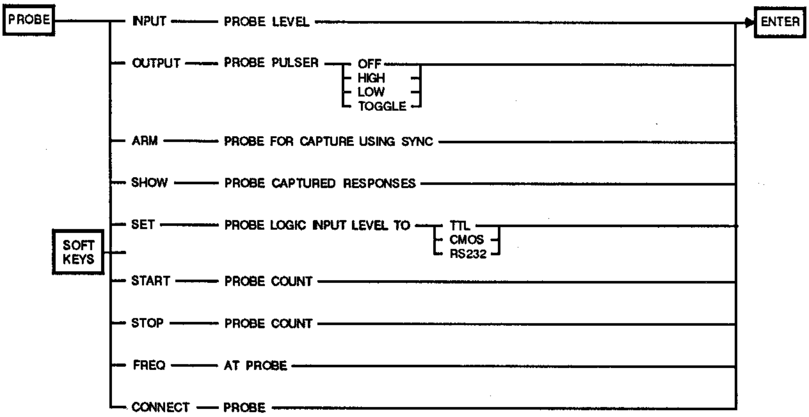
<pod-dependent choices>

An example of an available test is the QWK_RD (quick-read) test performed by the Z8000 pod. To select a test, press the corresponding softkey and press ENTER. At this point, you may be prompted for parameters; the pod then runs the test.

The pod operator's manual for your pod describes the tests available. For a list of pod tests for a particular pod, refer to the *Supplemental Pod Information for 9100A/9105A Users Manual.*

PROBE 5.17.

Causes outputs to and inputs from the UUT. Information flows through the probe during times set by the SYNC command.



Probe Capabilities 5.17.1.

The probe can perform two types of functions:

- Input (from the UUT): The probe can gather the following types of responses from the UUT:
  - Present logic levels (INPUT). Displayed as high (1), low (0), or invalid (X) on the operator's display.
  - Asynchronous level histories (INPUT, or ARM followed by SHOW). Displays all levels observed at the probe tip since the last execution of a command affecting the probe.
  - Clocked level histories (INPUT, or ARM followed by SHOW). Displays levels observed at all clock edges at the probe tip since the last command affecting the probe.
  - CRC signatures (ARM followed by SHOW).
  - Input frequency (FREQ). Frequency of rising transitions at the probe tip. Rising transitions are defined below.
  - Transition counts (START/STOP COUNT and ARM /SHOW). Rising transitions are counted at the probe tip. Rising transitions are low-to-high and invalid-to-high transitions.
- Output (to the UUT): The probe can output a series of pulses. The pulses can be high, low, or can toggle (alternate) between high and low.

Probe Logic-Level Lights 5.17.2.

There are three lights on the probe, which use the following colors to indicate logic levels:

- Red: logic 1 (high).
- Yellow: logic X (high-impedance or invalid).
- Green: logic 0 (low).

The information conveyed by the lights depends on the current SYNC command:

- **SYNC PROBE TO FREERUN CLOCK:** The yellow light's "X" sensing is filtered as described below. All the lights show the current static level. If the input changes slowly, the lights follow the input. If the input is a very short pulse, the lights would follow it too quickly for the human eye. In such cases, the pulse is "stretched" out long enough for observation. If the input is a wave that changes faster than the pulse-stretcher, such as a square wave, the red and green lights remain lit.
- **All other SYNC modes:** the yellow light's "X" sensing is unfiltered. The next section describes filtering. All the lights show the input level at the last clock edge.

The yellow light's "X" sensing operation (filtered or unfiltered) depends on the current SYNC mode, as explained in the previous section.

If filtered, an input must stay at the X level for a pre-defined amount of time before the yellow light comes on. This time is 100 nanoseconds for TTL/CMOS and 2 microseconds for RS-232 thresholds. Filtering avoids having an "X" state indicated at fast 0-to-1 or 1-to-0 signal transitions. Such transitions briefly pass through the X (invalid) region but reporting the X would be pointless.

Probe Synchronization 5.17.3.

Figure 2-8 showed how the probe interfaces with its external (clock module) control lines, the system, the internal PodSync line, and the UUT. Information can flow between the UUT and the system, through the probe, one bit at a time.

All PROBE commands described in this section except INPUT, FREQ, START COUNT and STOP COUNT are enabled and clocked during times determined by the SYNC command. The SYNC command synchronizes input or output to:

- **The system freerun clock.** This mode should only be used to synchronize the probe pulser output to the system's internal 1 kHz clock. Output begins at the first valid clock edge after the OUTPUT PROBE PULSER command is entered.
- **The pod's PodSync line.** Output or response gathering occurs at the trailing (rising) edge of the active PodSync signal.
- **The external (clock module) control lines.** The START, STOP, ENABLE, and CLOCK lines are used to specify when probe output or input occur.

  Start, Stop, and Enable conditions are defined using the SYNC PROBE TO EXT command. The COMMON line is always connected to UUT common. The CLOCK line, which is qualified by Start and an Enable condition, is connected to the desired UUT clock.
- **Internal.** This mode should only be used under control of a TL/1 program, which can generate an internal clock signal for synchronization.

CAUTION

Synchronization of OUTPUT PROBE differs slightly from that for all other PROBE and I/O MOD input and output. Timing diagrams are available in the SYNC key section.

INPUT PROBE LEVEL

Displays the present level at the probe tip, as 1 (high), 0 (low), or X (invalid). This level is measured at the time the command is entered.

Also displays the asynchronous level history (all input states seen) since this command was last entered. "X"-level sensing in the level history is filtered in order not to report an "X" at rapid signal transitions of low-to-high or high-to-low.

When filtered, an "X" input must stay at that level for a predefined amount of time before being recorded as such. This time is 100 nanoseconds for TTL/CMOS and 2 microseconds for RS-232 thresholds.

The INPUT command is not affected by the SYNC command.

OUTPUT PROBE PULSER OFF/HIGH/LOW/TOGGLE

The probe outputs a series of pulses which can be high, low, or alternating (toggling) between high and low levels. The OUTPUT PROBE PULSER OFF command shuts off the pulser.

See the SYNC section of the "Keypad Reference" section of this manual for timing details of the probe output pulser.

With the pulse threshold set to RS-232, a low is pulsed only to 0 V therefore, the green light will not come on (green < -3.2 V).

Note that when the probe threshold is set to RS-232, a low level is pulsed to 0 volts (see Appendix E). Since this is not a valid RS-232 low level, the green light of the probe will not be illuminated.

ARM PROBE FOR CAPTURE USING SYNC

Resets and arms (activates) the probe for response gathering.

The following responses are gathered after the command is entered at times specified by SYNC PROBE:

- CRC signatures.
- Transition counts.
- Clocked level history
- Asynchronous level history.

After entering this command, execute a manual or programmed stimulus and then enter SHOW PROBE to display resulting responses. A stimulus is a series of commands designed to generate known responses.

SHOW PROBE CAPTURED RESPONSES

Displays the responses gathered after ARM PROBE was entered. If the response-gathering period has not ended, an error message is displayed.

CAUTION

After entering ARM PROBE, a subsequent SHOW PROBE will display invalid responses if any of the *following commands were entered after ARM and before SHOW:*

PROBE command: SET, START, STOP, FREQ, or INPUT.

SYNC command: SYNC PROBE.

SET PROBE LOGIC INPUT LEVEL TO TTL/ CMOS/RS232

Specifies the probe input voltage threshold level. Setting this level also sets the time for which an X (invalid) input level has to be sensed before being recorded as such. Filtered in this manner is the "X" sensing of:

- The yellow probe light in SYNC PROBE TO FREERUN CLOCK mode.
- The asynchronous level history displayed upon an INPUT PROBE LEVEL command.

START PROBE COUNT

Starts counting rising transitions at the probe tip. Rising transitions are low-to-high or invalid-to-high. This command is not influenced by SYNC.

CAUTION

After pressing the START softkey, do not press it again before pressing the STOP softkey. Doing so would (incorrectly) reset the count to zero.

START PROBE COUNT and STOP PROBE COUNT are an asynchronous counting operation unaffected by SYNC.

STOP PROBE COUNT

Stops counting rising transitions at the probe tip and displays the count. This command is not influenced by SYNC.

FREQ AT PROBE

Calculates rising-transition frequency at the probe tip and displays the frequency. This command is not influenced by SYNC.

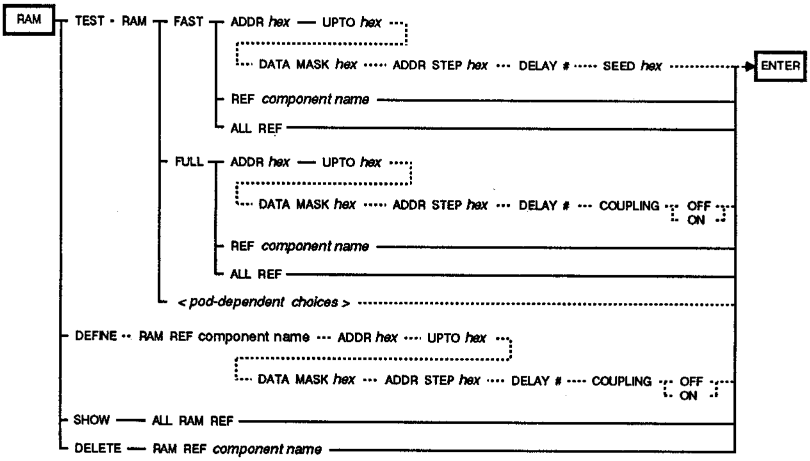
CONNECT PROBE

The display shows how the external clock module control lines should be connected to component pins on the UUT. The information comes from the test program currently being run. The operator should connect the lines as indicated. An asterisk (*) on the display indicates that there has been "no change"; that is, the control line was connected to the same UUT component pin as of the last time the system prompted the operator to make a connection with that line. The operator should verify that the lines are connected as indicated. If all the connections are the same (none have changed since the last connect instruction), the connect command will not prompt the operator.

The lines are named START, STOP, ENABLE, CLOCK, and COMMON. The response-gathering parameters for these lines are set using SYNC.

RAM 5.18.

Tests RAM components at a specified address range looking for cells that are stuck, dynamically coupled, or aliased. Also tests for address or data lines that are tied together or stuck at one level. OPTION fields should always be checked before pressing ENTER.


RAM
TEST · RAM
FAST
ADDR hex — UPTO hex
DATA MASK hex ····· ADDR STEP hex ··· DELAY # ····· SEED hex
ENTER
REF component name
ALL REF
FULL
ADDR hex — UPTO hex
DATA MASK hex ····· ADDR STEP hex ··· DELAY # ··· COUPLING OFF ON
REF component name
ALL REF
< pod-dependent choices >
DEFINE ·· RAM REF component name ··· ADDR hex ···· UPTO hex
DATA MASK hex ··· ADDR STEP hex ···· DELAY # ···· COUPLING OFF ON
SHOW —— ALL RAM REF
DELETE — RAM REF component name

There are three types of RAM tests:

- FAST: A short test performed by the system.
- FULL: A complete test performed by the system.
- pod-dependent: Tests performed by the pod.

You can name each component making up the RAM, defining it by a name and address range. Doing this reduces the keystrokes needed for later TEST commands. The definition can also be stored on a user disk.

TEST RAM ...

FAST ...

Performs a probabilistic RAM test: one based on a random yet thorough sampling of a given range. The test is fast, performing only 5 accesses at each location in the range. All data patterns are written to the RAM with equal probability. The test detects stuck RAM cells and address or data lines that are open or stuck.

The test also finds the following faults (probability of detection given):

- Internal RAM faults that affect an entire row or column (better than 0.999999).
- Aliasing between data bits at the same address (better than 0.5).

TEST RAM ...

FAST ...

(continued)

ADDR <hex> UPTO <hex> DATA MASK <hex> ADDR STEP <hex> DELAY # SEED <hex>

Tests the masked data bits over the specified address range. DATA MASK specifies which data bits to test. ADDR STEP sets the increments between addresses.

DELAY # specifies the number of milliseconds between memory accesses - a useful feature for testing dynamic RAM, which must be periodically refreshed.

This test writes a sequence of randomly generated data words to memory. The SEED value determines how the data is generated. If SEED is zero, the sequence of random data words is different at each pass through the memory. This procedure is always recommended. If SEED is not zero, for a given SEED value, the sequence of random data words is the same for each pass through the memory.

REF <component name>

Performs a probabilistic (fast) test for a component of the named type using parameters previously defined for that component using the DEFINE softkey and the OPTION command. This is a way to avoid repetitively entering the TEST RAM FAST ADDR command.

TEST RAM ...

FAST ...
(continued)

ALL REF

Performs a probabilistic (fast) RAM test for all components whose parameters have been defined (see DEFINE on the following pages).

FULL ...

Performs a deterministic (complete) test of RAM functionality for the specified address range.

The test combines two algorithms: a test for static or dynamic coupling of memory cells in the same data word, and the "Suk and Reddy B-Test." The latter is a standard algorithm for memory testing that typically makes 20 passes over the RAM.

ADDR <hex> UPTO <hex> DATA MASK <hex> ADDR STEP <hex> DELAY # COUPLING OFF/ON

DATA MASK specifies the data bits to test. ADDR STEP sets increments between addresses. DELAY # specifies the milliseconds between memory accesses: a useful feature for testing dynamic RAM, which must be periodically refreshed to retain data.

If the COUPLING option is ON, the RAM is tested for static or dynamic coupling of memory cells within one data word. Most words are formed by RAM components that are only one cell wide; in such cases, coupling is unlikely, and the test time can therefore be reduced by turning the COUPLING option OFF.

TEST RAM ...

FULL ...
(continued)

REF <component name>

Performs a deterministic test for a component of the named type, using parameters previously defined by DEFINE RAM. This is a way to avoid repetitively entering the TEST RAM FULL ADDR command.

ALL REF

Performs a deterministic test for all RAM components whose parameters were previously defined using the DEFINE softkey.

<pod-dependent choices>

The pod microprocessor performs a very fast RAM test (if available), independently of the 9100A/9105A system. An example is the Z8000's TEST RAM QUICK test. Refer to your pod manual for details. Pod-resident tests indicate that a fault has been found but do not indicate a probable cause.

DEFINE RAM REF <component name> ADDR <hex> UPTO <hex> DATA MASK <hex> ADDR STEP <hex> DELAY # SEED <hex> COUPLING OFF/ON

Defines values of the address range, mask, address increment, delay, and coupling for RAM tests (FULL or FAST) of the named component. The parameters are explained in the TEST RAM FAST section.

DEFINE RAM REF <component name> ADDR <hex> UPTO <hex> DATA MASK <hex> ADDR STEP <hex> DELAY # SEED <hex> COUPLING OFF/ON
(continued)

Before completing this command by pressing ENTER, select the appropriate OPTION fields, which are stored along with the other defined values.

The mask indicates the data bits to be tested. For example, a mask of FF00 (binary 1111 1111 0000 0000) specifies that only the higher-order 8 bits of a 16-bit data bus are to be tested.

COUPLING is ignored by TEST RAM FAST. SEED is ignored by TEST RAM FULL.

After defining parameters for a named RAM and selecting appropriate OPTION fields, you can store the information for later use. Press the SETUP MENU key, press the SAVE softkey and press ENTER to store the parameters on disk. DELETE RAM deletes them.

When SETUP SAVE (SETUP MENU command) is executed, address options specified along with DEFINE RAM are saved with the other parameters.

SHOW ALL RAM REF

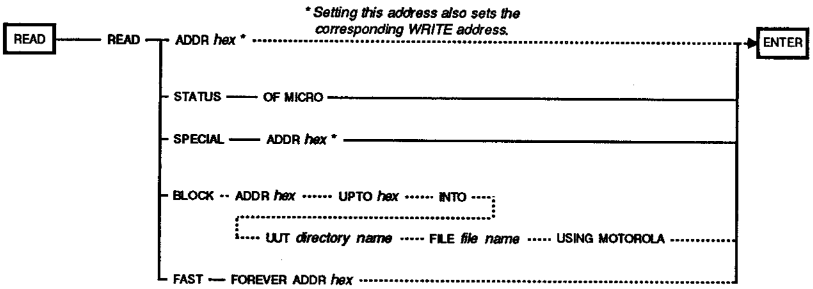
Displays the values of the address range, mask, address increment, delay, seed, and coupling for all components for which these values have been defined.

DELETE RAM REF <component name>

Erases the values of the address range, mask, address increment, delay, seed, and coupling associated with a component name. Do this when you wish to redefine the reference.

READ 5.19.

Displays data from memory, I/O, or other addresses and writes the data into a file, if necessary.



You can read:

- A UUT memory location or range of locations. This data can be written to a file. The read cycle can be observed on an oscilloscope.
- Pod status lines.
- A pod memory address, for some pods.
- An I/O address.

READ ...

ADDR <hex>

Displays data located at the specified address.

STATUS OF MICRO

Displays conditions of the pod status lines without performing a pod operation. Status data bit patterns are displayed as a hex number. The line corresponding to each bit is described in the pod manual or on the pod decal.

SPECIAL ADDR <hex>

Displays the data at the specified UUT virtual address or pod memory address. This command is pod dependent. It should only be used to access the results of quick pod-resident tests, because it causes the system to lose some fault-reporting capability. Refer to the pod manual for details on the use of this command.

The command can be used to simulate 9010 pods, which use virtual addresses to distinguish between I/O and memory addresses, between bytes and words, between user memory and program memory, and so on. In the 9100A/9105A, the OPTION key is used for such purposes.

READ ...

BLOCK ADDR <hex> UPTO <hex> INTO UUT <directory name> FILE <file name> USING MOTOROLA

Reads the data located in the specified address range and writes it into a file using the specified format. Information about the starting address and number of data bits read is also written to the file.

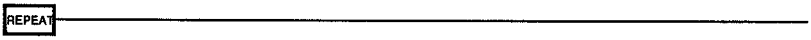
You can use this command to read a program into a file and then use the WRITE BLOCK command to load the file into UUT memory in preparation for a RUN UUT command.

FAST FOREVER ADDR <hex>

Command execution begins with one read operation at the specified address. This is followed by a very rapid sequence of reads at the same address. The data read is not displayed. Connecting an oscilloscope's external trigger input to the 9100A/9105A TRIGGER OUTPUT triggers the oscilloscope at each read cycle.

REPEAT 5.20.

Repeats the last pod test, probe, or I/O module command executed.



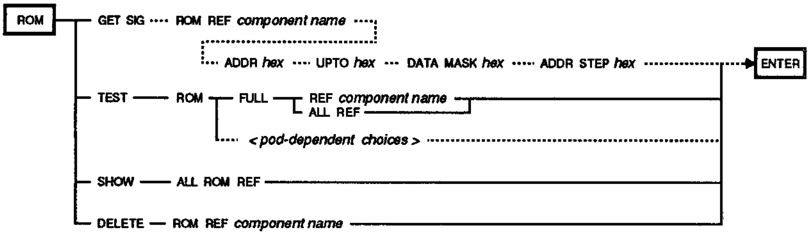
RESET 5.21.

Resets hardware and all fields to power-up default values. GFI (SUMMARY and SUGGEST) information is retained. All other information entered into fields since power-up or the last reset is lost, unless it was saved on disk using SETUP SAVE (SETUP MENU command).

RESET

ROM 5.22.

Obtains a signature for a particular address range of a ROM component and tests it by comparing the signature to a previously stored signature.



If the tested signature does not match the stored signature, a diagnostic routine looks for address or data lines that are open, tied, or stuck (high or low).

You should store signatures from a known-good ROM under the "component name" reference. They can also be stored on disk using the SETUP MENU key.

GET SIG ROM REF <component name> ADDR <hex> UPTO <hex> DATA MASK <hex> ADDR STEP <hex>

Gathers a CRC signature for a ROM on a known-good UUT. You should name the component and specify the address range and increment for which to collect a signature.

Masking allows you to be selective about the signature gathered. For example, if the UUT has a 16-bit data bus and two 8-bit ROM components, you can gather a signature for just one component by masking half the bus—the half written to by the other component. Bit positions corresponding to "1" in MASK are tested.

After defining the parameters for a ROM you can obtain signatures for all similarly defined components using RAM REF. To save the signature from a known-good ROM on disk, use SETUP SAVE (SETUP MENU key). When SETUP SAVE is executed, address options specified with DEFINE RAM are saved along with the other parameters.

TEST ROM ...

FULL ...

Tests the ROM with the system's full test. Obtains a CRC signature from the ROM under test and compares it to a previously stored CRC signature from an identical ROM that is known to work correctly. Signature properties guarantee detection of any one-bit fault and any single sequence of 16 or fewer incorrect bits.

If the data in the ROM is distributed at random, the probability of a faulty ROM producing a valid signature is less than 0.0001.

TEST ROM ...

FULL ...
(continued)

If the ROM signature is found to be incorrect, a diagnostic routine uses all signature information to look for address and data lines that are open, or tied to each other, or stuck high or stuck low.

REF component name

Compares the CRC signature of the specified ROM component to the CRC signature stored using GET SIG ROM.

ALL REF

Verifies the CRC signatures of all known-good ROM components from which signatures have been gathered and stored.

<pod-dependent choices>

Performs a very fast ROM test. The test is carried out by the pod microprocessor, independently of the 9100A/9105A system. An example is the TEST ROM QUICK command for the Z8000 pod. The pod-resident tests indicate that a fault has been found but do not indicate the probable cause.

SHOW ALL ROM REF

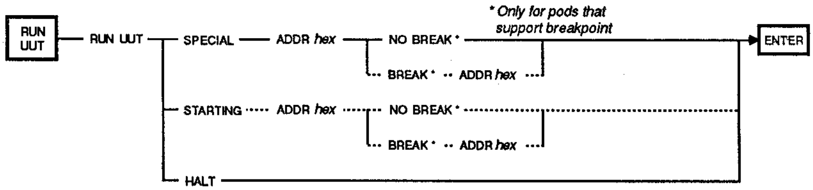
Displays the signatures and values of address range, mask, and address increment for all ROM components for which a signature has been gathered and stored using GET SIG ROM. The stored signatures are typically from a known-good UUT.

DELETE ROM REF <component name>

Erases the stored signature and values of address range, mask, and address increment for the specified ROM component.

RUN UUT 5.23.

Transfers control of the UUT to the pod microprocessor, which executes program instructions from UUT memory. In this mode, the UUT is not controlled by the 9100A/9105A system. Reset the UUT with its own reset switch before entering this command.



RUN UUT ...

SPECIAL ADDR <hex> ...

Causes the pod to begin executing UUT instructions from UUT memory at the specified special address independently of the 9100A/9105A. This command is nearly always used for RUN UUT tests. The default ADDR value is pod specific and should not be changed. Control reverts to the system if a RUN UUT HALT or a RESET command is entered.

NO BREAK

A softkey option in pods such as the 80286. The microprocessor executes UUT instructions until the RUN UUT HALT or RESET command is entered.

BREAK ADDR <hex>

A softkey option in pods such as the 80286. The microprocessor executes UUT instructions until it reaches the instruction at the specified memory address. The pod does not execute that instruction. Execution can also be stopped by the RUN UUT HALT or RESET command.

STARTING ADDR <hex> NO BREAK/BREAK

This command differs from SPECIAL ADDR <hex> NO BREAK/BREAK only in that:

- It is not normally used in RUN UUT tests. Its default value is zero.
- STARTING ADDR is not a special address.
- RUN UUT STARTING has no address options.

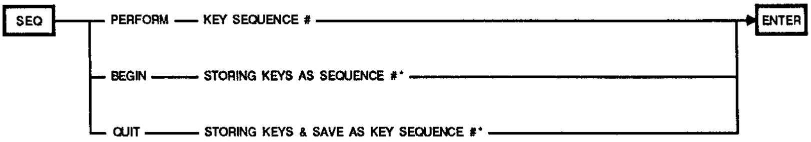
RUN UUT ...

HALT

This command is a convenient way to stop the pod's RUN UUT operation returning control of the UUT to the 9100A/9105A. RUN UUT HALT is also a convenient way of resetting only the pod.

SEQ 5.24.

Allows the operator to store a sequence of up to 99 keystrokes. Up to ten sequences can be defined (numbered from 0 through 9). This sequence number is used to identify a sequence when it is stored or executed.


SEQ
PERFORM
KEY SEQUENCE #
BEGIN
STORING KEYS AS SEQUENCE #*
QUIT
STORING KEYS & SAVE AS KEY SEQUENCE #*
ENTER

** Whatever you enter after pressing ENTER is recorded as the key sequence, until you press the SEQ key, the QUIT softkey, and the ENTER key, in that order.*

To start storing a keystroke sequence, press SEQ, press the BEGIN softkey, give the sequence a number, press ENTER, and then type the keystrokes you want stored.

To stop storing keystrokes, press SEQ again, press the QUIT softkey, check that the sequence number is unchanged, and press ENTER. Storage also stops if an error or a UUT fault is encountered.

To reproduce the sequence, press SEQ, press the PERFORM softkey, type the sequence number, and press ENTER.

CAUTION

Keystrokes are reproduced by PERFORM exactly as they were entered after BEGIN. When a command is executed during PERFORM, its fields may differ in value from what they were during the BEGIN STORING sequence.

To avoid any ambiguity, enter commands during the BEGIN STORING sequence as follows:

1. Press the key once to display the command.

2. Move the cursor to all relevant fields and specify the value of each field. This must be done even if the field is already specified as desired.

3. Press the ENTER key if needed for the command.

Remember that the system configuration as specified by the SETUP MENU command should be the same during the PERFORM and BEGIN operations. Where necessary, the system calibration and self-tests should be performed before the PERFORM operation. Self-tests, calibration, and configuration are explained in Section 4.

PERFORM KEY SEQUENCE

Executes the key sequence identified by the number.

Ending a sequence with the PERFORM softkey will cause the sequence to execute and to loop until the STOP key is pressed.

BEGIN STORING KEYS AS SEQUENCE

Starts key sequence storage. Type a number from 0 to 9 to identify the sequence. Keys typed after this command is entered are stored.

QUIT STORING KEYS & SAVE AS KEY SEQUENCE

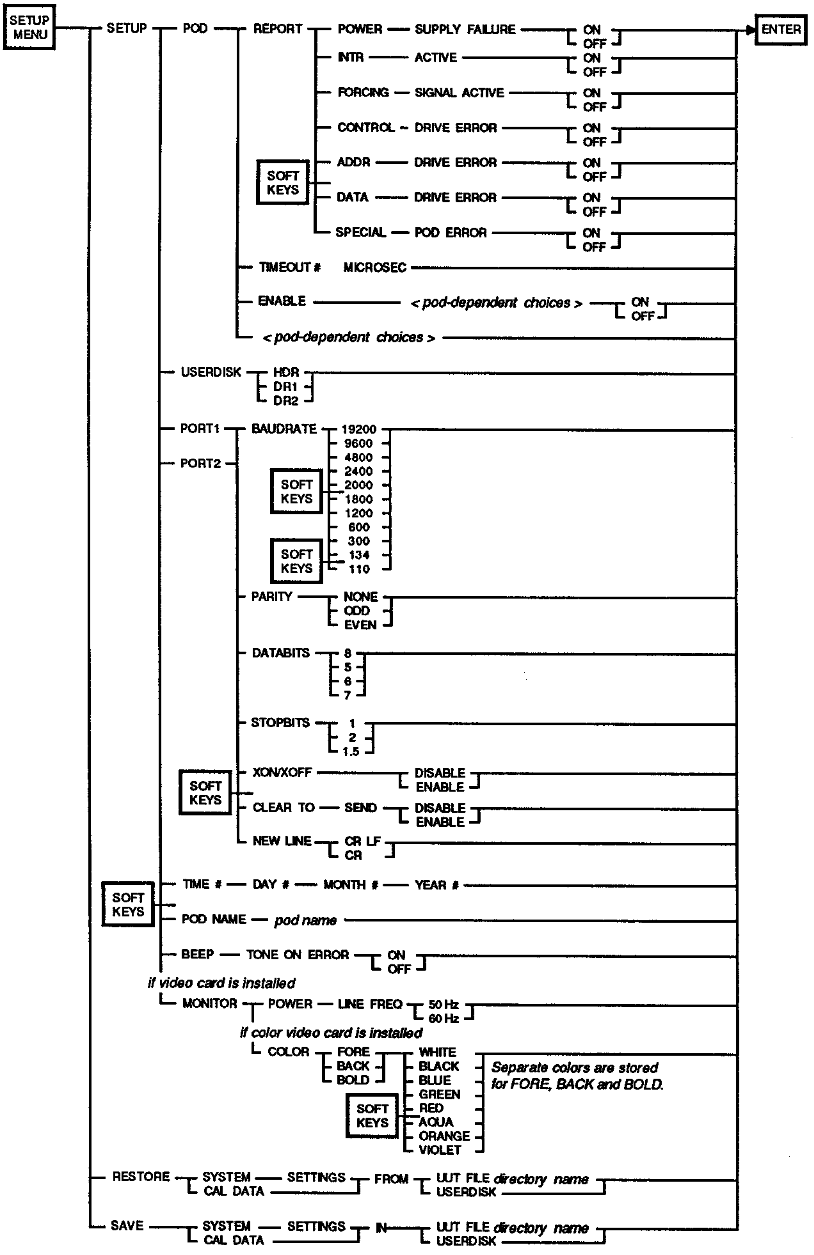
Stops key sequence storage. The sequence is stored and identified by the number specified.

SETUP MENU 5.25.

Sets system variables such as communication port configurations, the internal real-time clock, and pod parameters.

Once you configure the system a certain way, you can store the settings on disk and recall them later. You can also store and recall certain I/O MOD, PROBE, RAM, ROM, and SYNC command settings.

SETUP MENU Key


SETUP MENU
SETUP
POD
REPORT
POWER — SUPPLY FAILURE — ON OFF
INTR — ACTIVE — ON OFF
FORCING — SIGNAL ACTIVE — ON OFF
CONTROL – DRIVE ERROR — ON OFF
ADDR — DRIVE ERROR — ON OFF
DATA — DRIVE ERROR — ON OFF
SPECIAL — POD ERROR — ON OFF
SOFT KEYS
TIMEOUT # MICROSEC
ENABLE — <pod-dependent choices> — ON OFF
<pod-dependent choices>
USERDISK — HDR DR1 DR2
PORT1 — BAUDRATE — 19200 9600 4800 2400 2000 1800 1200 600 300 134 110
PORT2
SOFT KEYS
SOFT KEYS
PARITY — NONE ODD EVEN
DATABITS — 8 5 6 7
STOPBITS — 1 2 1.5
XON/XOFF — DISABLE ENABLE
CLEAR TO — SEND — DISABLE ENABLE
NEW LINE — CR LF CR
SOFT KEYS
TIME # — DAY # — MONTH # — YEAR #
POD NAME — pod name
BEEP — TONE ON ERROR — ON OFF
SOFT KEYS
if video card is installed
MONITOR — POWER — LINE FREQ — 50 Hz 60 Hz
if color video card is installed
COLOR — FORE BACK BOLD — WHITE BLACK BLUE GREEN RED AQUA ORANGE VIOLET
SOFT KEYS
Separate colors are stored for FORE, BACK and BOLD.
RESTORE — SYSTEM — SETTINGS / CAL DATA — FROM — UUT FILE directory name / USERDISK
SAVE — SYSTEM — SETTINGS / CAL DATA — IN — UUT FILE directory name / USERDISK
ENTER

SETUP ...

POD ...

REPORT ...

Determines whether or not the system indicates faults discovered by the pod. Note that the messages are the same as for BUS test errors. However, you cannot disable the fault message display for the BUS test.

POWER SUPPLY FAILURE ON/OFF

Specifies whether or not to report a faulty UUT power supply.

INTR ACTIVE ON/OFF

Specifies whether or not to report an active UUT interrupt line.

FORCING SIGNAL ACTIVE ON/OFF

Specifies whether or not to report an active UUT forcing line.

CONTROL DRIVE ERROR ON/OFF

Specifies whether or not to report that a UUT control line cannot be driven.

SETUP ...

POD ...

REPORT ...
(continued)

ADDR DRIVE ERROR ON/OFF

Specifies whether or not to report that a UUT address line cannot be driven.

DATA DRIVE ERROR ON/OFF

Specifies whether or not to report that a UUT data line cannot be driven.

SPECIAL POD ERROR ON/OFF

Specifies whether or not to report that a special pod error has occurred.

TIMEOUT # MICROSEC

Specifies the maximum number of microseconds the system waits for the pod to respond to a command. The number can be from 0 through 9,999,999.

ENABLE <pod-dependent choices> ON/OFF

Enables or disables lines specific to a particular pod. If a line is disabled, the pod microprocessor will not respond to the corresponding signal.

SETUP ...

POD...
(continued)

<pod-dependent choices>

Other pod-dependent set up features. Refer to your pod manual for details.

USERDISK HDR/DR1/DR2

Specifies the user disk drive. The system uses this disk for information such as the GFI database, system configurations, and calibration data. The softkeys labels shown are system specific.

PORT1/PORT2 ...

This command sets communication parameters for serial ports 1 and 2. The pins used are described in the table below.

Pin	*Name*	*Function*
2	Tx	Transmit data
3	Rx	Receive data
4	RTS	Request to send, asserted always
5	CTS	Clear to send
7	signal gnd	Common (port1) or ground (port2)
20	DTR	Data terminal ready

Port 1 is not earth referenced. It should be used if a port is needed to test UUTs. Port 2 is earth referenced and should be used to connect to a host computer, modem, or printer.

SETUP ...

PORT1/PORT2 ...
(continued)

BAUDRATE <b>

Sets the rate (bits per second) at which the system transmits or receives information. "b" represents the following softkey selectable baud rates: 19200, 9600, 4800, 2400, 2000, 1800, 1200, 600, 300, 134, and 110.

PARITY NONE/ODD/EVEN

Specifies how to calculate the parity bit transmitted for the data information. Parity bits are also present in data received by the system, which uses the same method to check that the data is accurate.

DATABITS 8/5/6/7

Sets the number of data bits in the data portion of information transmitted or received by the system.

STOPBITS 1/2/1.5

Sets the number of stop bits in a byte of data transmitted or received by the system.

SETUP ...

PORT1/PORT2 ...

(continued)

XON/XOFF DISABLE/ENABLE

Enables or disables software handshaking using XOFF (Ctrl-S, hex 13) and XON (Ctrl-Q, hex 11). Handshaking typically occurs between the system and a host computer.

If the system receives XOFF, it will stop transmitting data. The system may also send XOFF to the host to indicate that the system needs to stop receiving data to avoid overrunning its input buffer.

If the system receives XON, it will resume transmitting data. The system may also send XON to the host to indicate that data transmission to the system can resume.

CLEAR TO SEND DISABLE/ENABLE

Enables or disables hardware handshaking. CTS (clear-to-send) is sensed by the system, enabling data transmission by the system if true. When not true, CTS disables transmission. RTS (request-to-send) is always asserted true by the system.

NEWLINE CRLF/CR

Sets the character recognized as the end of a line. CR is a carriage return (hex 0D). CRLF is a carriage return (hex 0D) followed by a linefeed (hex 0A).

SETUP ...
(continued)

TIME # DAY # MONTH # YEAR

Displays the present time, day, month, and year. The display does not change with time. To reset the time, change the fields. The new values are effective when the ENTER key is pressed.

The 9100A's built-in real-time clock is set to the values entered, which are retained when the system is powered down. The real-time clock is an option on the 9105A. The TIME command works even on a 9105A without this option, but the values entered are lost when the system is powered down.

POD NAME <pod name>

Specifies the microprocessor being used, in cases where the pod can support more than one. This information can be found in Appendix C "Pod-Related Information." For example, specify "Z8001" for a Z8000 pod using a Z8001 microprocessor. When a new pod is installed, the new pod name is specified automatically if RESET is pressed.

BEEP TONE ON ERROR ON/OFF

Turns the system beep generation on or off.

MONITOR

If the 9100A/9105A has a monitor video control card, this softkey option appears.

SETUP ...

MONITOR ...

(continued)

POWER LINE FREQ 60HZ/50HZ

Sets the vertical scan rate on the video control card to match the local power line frequency (60 or 50 Hz).

COLOR

If the 9100A/9105A has a color monitor video control card, this softkey option appears.

FORE WHITE/BLACK/BLUE/ GREEN/RED/AQUA/ORANGE/ VIOLET

Specifies foreground color for a color monitor.

BACK WHITE/BLACK/BLUE/ GREEN/RED/AQUA/ORANGE/ VIOLET

Specifies background color for a color monitor.

BOLD WHITE/BLACK/BLUE/ GREEN/RED/AQUA/ORANGE/ VIOLET

Specifies the color with which to highlight boldface characters.

RESTORE ...

SYSTEM SETTINGS FROM UUT FILE <directory name>/USERDISK

Configures the system by restoring all previously stored system settings from the UUT directory.

If there is no UUT directory, press the USERDISK softkey to restore settings from the current userdisk. Restoring saved settings will overwrite all present settings. Settings are specified by:

- SETUP MENU.
- SET PROBE.
- SET I/O MOD.
- DEFINE RAM REF.
- GET SIG ROM REF.
- SYNC.
- SEQ.

When you are prompted for the UUT directory name, pressing the HELP key will display a list of available UUT directory names.

CALDATA FROM UUT FILE <directory name>/USERDISK

Restores calibration data from the UUT directory or the userdisk directory. The data was saved previously using SETUP MENU (SAVE CAL).

If there is no directory, press the USERDISK softkey to restore data from the current userdisk. Restoring saved data will overwrite all present settings.

RESTORE ...

CALDATA FROM UUT FILE <directory name>/USERDISK

(continued)

CAUTION

SETUP RESTORE CAL DATA is not a substitute for calibration. You should carefully read Section 4, which describes when calibration is necessary.

SAVE ...

SYSTEM SETTINGS IN UUT FILE <directory name>/USERDISK

Saves system configurations (all system settings) in the UUT directory. If no directory exists press the USERDISK softkey to save the settings on the current userdisk.

Use this command to save settings from the SETUP MENU key, SET I/O MOD, SET PROBE, DEFINE RAM REF, GET SIG ROM, and the SYNC key.

When you are prompted for a UUT directory name, pressing the HELP key will display a list of available UUT directory names.

CALDATA IN UUT FILE <directory name>/ USERDISK

Saves data generated by calibration in the UUT directory. If no directory exists press the USERDISK softkey to save the data on the current userdisk.

Use this command to save data generated by the following MAIN MENU commands:

- CAL PROBE TO EXT.
- CAL PROBE TO POD.
- CAL I/O MOD TO EXT.
- CAL I/O MOD TO POD.

When you are prompted for a UUT directory name, pressing the HELP key will display a list of available UUT directory names.

SOFT KEYS 5.26.

Scrolls hidden sets of softkey labels into view if the MORE SOFTKEYS light is on.

SOFT KEYS

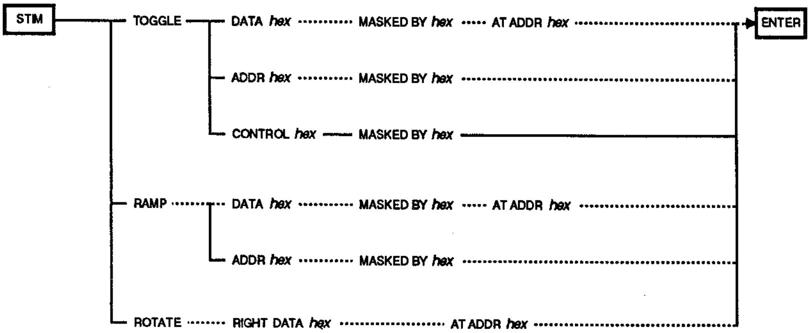
A set of up to five softkey labels can be displayed at one time. If a field has more than five softkey options, extra sets of labels are hidden from view. To select a softkey from a hidden set of labels, press the SOFT KEYS key to scroll the hidden set into view, and then press the desired softkey.

Hidden sets of labels are scrolled into view by pressing the SOFT KEYS key as often as necessary. After the last set is in view, repeatedly pressing this key then scrolls the labels back towards the first set.

If there are only two sets of labels, pressing this key would toggle alternate sets into view.

STIM 5.27.

Performs a series of READ or WRITE commands. These functions are used to provide stimuli to locate a fault or troubleshoot one that has already been found.



A data mask (hex number) is used to mask (identify) bit positions in the data being written or to mask bit positions in the address from which data is being read. Masked bits are those affected by the command to be executed. Masking is also used in other commands, such as RAM and ROM. An example of mask calculation is shown here:

	Hex Number	*Bit Pattern*
If the data is:	BD26	1011 1101 0010 0110
To test underlined bits, mask the data with:	0A10	0000 1010 0001 0000

To mask a bit in the data, the corresponding bit in the mask is set (logic 1). The other bits in the mask are cleared (logic 0). When a data bit is masked, its value does not affect the calculation of the mask.

TOGGLE ...

DATA <hex> MASKED BY <hex> AT ADDR <hex>

Performs two WRITE DATA operations for each set bit in the mask starting with the least-significant set bit. The mask is used to modify individual bits in data written to the specified address.

	Hex Number	*Bit Pattern*
Data specified	00A5	1010 0101
Mask the data with:	0049	0100 1001
The system writes the following data:	00A5	1010 0101
	00A4	1010 0100
	00A5	1010 0101
	00AD	1010 1101
	00A5	1010 0101
	00C5	1100 0101

Each pair of WRITE operations first outputs the original data word. Next, the word is output with one bit in it toggled to its opposite value.

The toggled bits correspond to set bits in the mask, beginning with the mask's least-significant set bit and progressing in order to the most-significant set bit.

TOGGLE ...
(continued)

ADDR <hex> MASKED BY <hex>

Performs two READ ADDR commands for each set bit in the mask starting with the least-significant set bit. The mask is used to modify individual bits in the address at which data is to be read.

	Hex Number	*Bit Pattern*
Address:	12A5	0001 0010 1010 0101
Mask data with:	0058	0000 0000 0101 1000
Data read from the following addresses:	12A5	0001 0010 1010 0101
	12AD	0001 0010 1010 1101
	12A5	0001 0010 1010 0101
	12B5	0001 0010 1011 0101
	12A5	0001 0010 1010 0101
	12E5	0001 0010 1110 0101

Each pair of READ operations first inputs data from the original address. Next, the address has one bit in it toggled to its opposite value.

The toggled bits correspond to set bits in the mask, starting with the least-significant set bit and progressing in order to the most-significant set bit.

TOGGLE ...
(continued)

CONTROL <hex> MASKED BY <hex>

Performs two WRITE CONTROL operations on each masked control line starting with the line at the mask's LSB position. The example shows what happens when one of two control lines is masked.

Toggling more than one control line at any time is generally a useless exercise.

	Hex Number	*Bit Pattern*
CONTROL output	02	0000 0010
Mask one line with:	02	0000 0010
The system exercises the line with the following output:	02	0000 0010
	00	0000 0000

The pod decal and the pod manual describe the control lines that correspond to each bit. The example below is for a Z80 pod. The "*" indicates which lines can be written to by the user. The preceding example toggled bit 1, which corresponds to the microprocessor HALT line.

Data Bit #	*Z80 Control Function*
7	MREQ-
6	M1-
5	WR-
4	RD-
3	IORQ-
2	RFSH-
1	*HALT-
0	*BUSAK-

RAMP ...

DATA <hex> MASKED BY <hex> AT ADDR <hex>

Performs a series of WRITE DATA commands, to a specified address, each with a different data value. The mask modifies the data written. All masked data bits change with each WRITE operation, but unmasked bits are not altered.

The data values are determined from the original data by setting the values of all the masked bits to all combinations of ones and zeros, beginning with all zeros and ending with all ones.

	Hex Number	*Bit Pattern*
Data specified:	F3	1111 0011
Masked by:	07	0000 0111
The system writes the following data:	F0	1111 0000
	F1	1111 0001
	F2	1111 0010
	F3	1111 0011
	F4	1111 0100
	F5	1111 0101
	F6	1111 0110
	F7	1111 0111

RAMP ...
(continued)

ADDR <hex> MASKED BY <hex>

Performs a series of READ ADDR commands, each at a different address. The mask modifies the address. All masked address bits change with each READ operation, but unmasked bits are not altered.

The addresses are determined from the original address by setting the values of all the masked bits to all combinations of ones and zeros, beginning with all zeros and ending with all ones.

	Hex Number	*Bit Pattern*
Address:	1234	0001 0010 0011 0100
Mask data with:	07	0000 0000 0000 0111
Data read at the following addresses:	1230	0001 0010 0011 0000
	1231	0001 0010 0011 0001
	1232	0001 0010 0011 0010
	1233	0001 0010 0011 0011
	1234	0001 0010 0011 0100
	1235	0001 0010 0011 0101
	1236	0001 0010 0011 0110
	1237	0001 0010 0011 0111

ROTATE RIGHT DATA <hex> AT ADDR <hex>

Performs a series of WRITE DATA commands. First, the data is written to the specified address. Then the data is rotated right, the right-most bit becoming the left-most one. The process is repeated as many times as there are data bits.

The example below shows how the sequence of data is written to address FFFFE, according to the command ROTATE RIGHT DATA 1234 AT ADDRESS FFFFE.

Hex Data Output	*Bit Pattern*
1234	0001 0010 0011 0100
091A	0000 1001 0001 1010
048D	0000 0100 1000 1101
8246	1000 0010 0100 0110
4123	0100 0001 0010 0011
A091	1010 0000 1001 0001
D048	1101 0000 0100 1000
6824	0110 1000 0010 0100
3412	0011 0100 0001 0010
1A09	0001 1010 0000 1001
8D04	1000 1101 0000 0100
4682	0100 0110 1000 0010
2341	0010 0011 0100 0001
91A0	1001 0001 1010 0000
48D0	0100 1000 1101 0000
2468	0010 0100 0110 1000

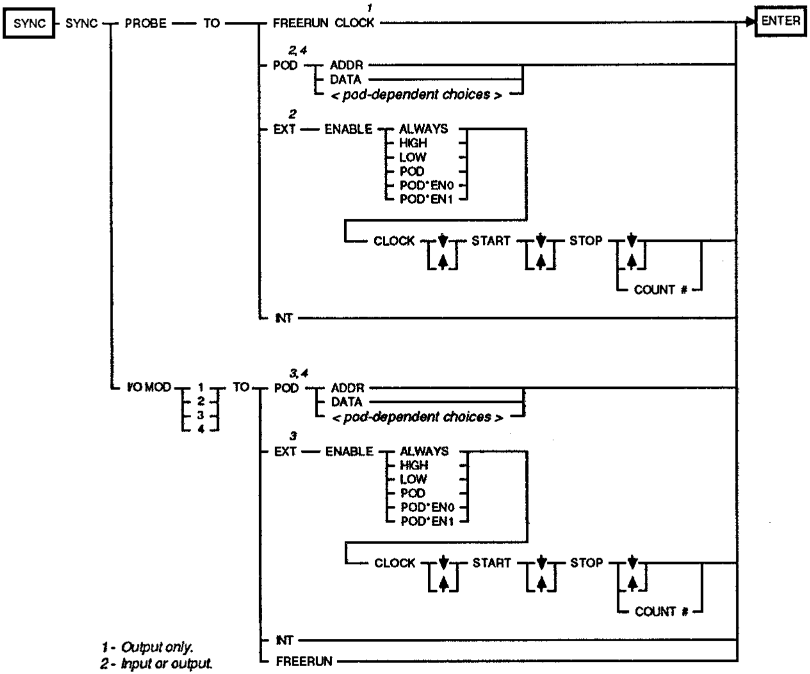
STOP 5.28.

Stops program execution or command execution. Execution can be resumed by pressing the CONT key.

STOP

SYNC 5.29.

Specifies the source of the clock signal to be used for the probe or I/O modules. For the probe, this clock is used to gather synchronous data, or to pulse synchronously, or both. For the I/O module, the sync mode is used to gather synchronous data only.



1 - Output only.
2 - Input or output.
3 - Input only.
4 - Setting the probe synchronization sets the corresponding I/O module synchronization, and vice versa.

For both the probe and I/O module, there are four sync modes.

- External Sync.
- Pod Sync.
- Freerun Sync.
- Internal Sync.

External Sync Mode 5.29.1.

This mode qualifies the external CLOCK line with the external START, STOP, and ENABLE lines. For the probe, the lines are available through the clock module. The I/O module has its own external lines.

Start, Stop, and Clock are edge-sensitive inputs from the UUT. Each can be made to respond to falling or rising edges. The sync period can also be programmed to end after a specified number of valid clock pulses, in which case the Stop input is ignored. Enable is a level sensitive signal and can be specified as valid HIGH, LOW, or ALWAYS. When the sync enabling condition is set to ALWAYS, the Enable input is ignored.

NOTE

The enabling condition need not always depend on the specified state (HIGH, LOW, ALWAYS) of the Enable input. The enabling condition can also involve the state of the internal PodSync line.

After an ARM (I/O MOD or PROBE) command is entered, and after a valid Start edge is detected, the sync measurement period begins. Start is recognized independently of the enabling condition, but data is only gathered after the enabling condition becomes true. Asynchronous data is gathered immediately after this point. Synchronous data is gathered after the same point, but only at the selected clock edges.

The data gathering period ends when the Stop condition becomes true: the selected Stop edge occurs, or a programmed number of clock edges completes. After the period ends, enter the appropriate SHOW (I/O MOD or PROBE) command, which displays data gathered by the following:

- **CRC signature registers:** Contain signature data clocked in at each enabled clock edge between Start and Stop.
- **Clocked level history register:** Contains all logic levels (1, 0, X) seen at each each enabled clock edge between Start and Stop.
- **Asynchronous level history register:** Contains all logic levels (1, 0, X) seen while enabled, between Start and Stop. An X level has to be present for a fixed period before being recorded as such. This is because an X should not be recorded during fast 0-to-1 or 1-to-0 transitions, which do pass briefly through the invalid (X) region.
- **Transition counter:** Records each rising transition at the input, between Start and Stop, while enabled. A rising transition is either 0-to-1 or X-to-1.

Pod Sync Mode 5.29.2.

This mode uses PodSync, an internal pod signal, as the clock. PodSync is generated by the pod at each READ and WRITE. Its generation can be made to depend on valid address, data, or other (pod-dependent) cycles. External lines (START, STOP, etc.) are ignored.

Data is gathered after an ARM (PROBE or I/O MOD) command. Synchronous data is gathered on the rising (trailing) edge of PodSync. Data is gathered until a SHOW (PROBE or I/O

MOD) command, which displays data gathered by the following:

- CRC signature registers.
- Clocked level history registers.
- Asynchronous level history registers.
- Transition counter.

The data is gathered as described previously for the "External Sync Mode," except that the CRC signatures and clocked level history are clocked by the rising PodSync edge.

Freerun Sync Mode 5.29.3.

In this mode, the probe uses the 9100A/9105A system's internal 1 kHz clock for asynchronous output. The external START, STOP, ENABLE, and CLOCK lines are ignored.

Although input through the probe is possible, this mode is designed only to clock the probe's pulsing output via the OUTPUT PROBE PULSER command. Asynchronous level history and transition count data can be gathered between ARM PROBE and SHOW PROBE commands. CRC signatures and clocked level histories are not gathered.

For the I/O module, this sync mode is more usefully invoked under TL/1 program control, rather than from the keypad.

Internal Sync Mode 5.29.4.

This sync mode is designed to use a clock signal generated internally by a program. It is therefore only useful when invoked under TL/1 program control. The external START, STOP, CLOCK, and ENABLE lines are ignored.

SYNC ...

PROBE TO ...

FREERUN CLOCK

Freerun sync mode. Data output to the probe is clocked by the system's internal freerun clock.

POD ...

Pod sync mode. Data input/output through the probe is synchronized to the pod's internal PodSync line. PodSync is defined as active by the conditions described below.

ADDR

This command specifies that PodSync is active during a valid pod address cycle. If the I/O module is set to a POD sync mode, entering this command also enters the SYNC I/O MOD TO POD ADDR command.

DATA

This command specifies that PodSync is active during a valid pod data cycle. If the I/O module is set to a POD sync mode, entering this command also enters the SYNC I/O MOD TO POD DATA command.

<pod-dependent choices>

This command specifies that if the specified, pod-dependent line is active, then PodSync is active during that time. If the I/O module is set to a POD sync mode, entering this command also enters the

SYNC ...

PROBE TO ...

<pod-dependent choices>

(continued)

SYNC I/O MOD TO <pod-dependent choice> command.

EXT ENABLE ... CLOCK ↓/↑ START ↓/↑ STOP ↓/↑/COUNT

External sync mode. Data input/output through the probe is synchronized and clocked by PodSync or the external measurement control lines from the clock module.

The ENABLE field is set to one of the softkeys (ALWAYS, HIGH, LOW, POD, POD*EN0 or POD*EN1) described below. Each softkey defines an enabling condition which must be true for Start to be enabled.

ALWAYS

Enabling condition is always true.

HIGH

Enabling condition is true if the ENABLE line is high.

LOW

Enabling condition is true if the ENABLE line is low.

SYNC ...

PROBE TO ...

EXT ENABLE ... CLOCK ↓/↑ START ↓/↑ STOP ↓/↑/COUNT

(continued)

POD

Enabling condition is true if the pod's PodSync line is active.

POD*EN0

Enabling condition is true if the pod's PodSync line is active and the ENABLE line is low.

POD*EN1

Enabling condition is true if the pod's PodSync line is active and the ENABLE line is high.

NOTE

Even when the probe is in external sync mode, an enabling condition involving POD will depend on the probe's earlier pod sync mode. For example, entering SYNC PROBE TO POD ADDR followed by SYNC PROBE TO EXT ENABLE POD will mean that the enabling condition (POD) is true during a valid pod address cycle.

SYNC ...

PROBE TO ...

EXT ENABLE ... CLOCK ↓/↑ START ↓/↑ STOP ↓/↑/COUNT

(continued)

The CLOCK field specifies whether data is clocked on a falling or rising clock signal edge.

The START field specifies whether data input/output begins on a falling or on a rising Start signal.

The STOP field specifies whether data input/output ends after a falling Stop signal, rising Stop signal, or after the number of clock pulses specified by COUNT. If COUNT is selected, the external STOP line is ignored.

COUNT Causes data input/output to end after the specified number of active clock edges, after the Start signal. The number can be from 1 to 65535.

INT

Internal sync mode. This mode is designed solely for use under TL/1 program control. The clock used for data input/output is generated by a TL/1 command.

SYNC ...
(continued)

I/O MOD <n> TO ...

POD ...

Pod sync mode. Data input through the specified I/O modules is synchronized to the pod's internal PodSync line. "n" is a softkey option: 1, 2, 3, or 4. PodSync is defined active by the conditions described below.

ADDR

This command specifies that PodSync is active during a valid pod address cycle. If the probe is set to a POD sync mode, entering this command also enters the SYNC PROBE TO POD ADDR command.

DATA

This command specifies that PodSync is active during a valid pod data cycle. If the probe is set to a POD sync mode, entering this command also enters the SYNC PROBE TO POD DATA command.

<pod-dependent choices>

This command specifies that if the specified, pod-dependent line is active, then PodSync is active during that time. If the probe is set to a POD sync mode, entering this command also enters the SYNC PROBE TO <pod-dependent choice> command.

SYNC ...

I/O MOD <n> TO ...

(continued)

EXT ENABLE ... CLOCK ↓/↑ START ↓/↑ STOP ↓/↑/COUNT

External sync mode. Data input through the specified I/O module is synchronized and clocked by PodSync or the I/O module's measurement control lines

ALWAYS

Enabling condition is always true.

HIGH

Enabling condition is true if the ENABLE line is high.

LOW

Enabling condition is true if the ENABLE line is low.

POD

Enabling condition is true if the PodSync line is active.

SYNC ...

I/O MOD <n> TO ...

EXT ENABLE ... CLOCK ↓/↑ START ↓/↑ STOP ↓/↑/COUNT

(continued)

NOTE

Even if the I/O module is in external sync mode, the enabling condition ENABLE POD depends on any earlier pod sync mode. For example, entering SYNC I/O MOD TO POD DATA followed by SYNC I/O MOD TO EXT ENABLE POD means that the enabling condition (POD) now becomes true during a valid pod data cycle.

The ENABLE field is set to one of the softkeys (ALWAYS, HIGH, LOW, or POD) described on the previous page. Each softkey defines a condition which must be true for Start to be enabled.

The CLOCK field specifies whether data is clocked on a falling or rising clock signal edges.

The START field specifies whether data input begins on a falling or a rising Start signal.

The STOP field specifies whether data input ends after a falling Stop signal, rising Stop signal, or after the number of clock pulses specified by COUNT. If COUNT is selected, the Stop signal is ignored.

COUNT causes data input to end after the specified number of active clock edges, after the Start signal. The number can be from 1 to 65535.

SYNC ...

I/O MOD <n> TO ...
(continued)

INT

Internal sync mode. This mode is designed solely for use under TL/1 program control. The clock used for data input/output is generated by a TL/1 command.

FREERUN CLOCK

Freerun sync mode I/O module input/output is totally asynchronous. This mode is intended for use only under TL/1 program control, but if selected, this mode will allow transition counts or asynchronous level histories to be recorded.

Sync Mode Examples 5.29.5.

The following examples show how you can use the probe and I/O modules for input/output in a variety of sync modes. The last two I/O module examples also show how the data gathered during the sync period is displayed.

Freerun Sync (Probe) Example

Only the probe should be synchronized to the system freerun clock, and only for output. The probe output pulses to the state specified by the OUTPUT PROBE PULSER command.

Figure 5-5 is an example of the timing for the command: SYNC PROBE TO FREERUN CLOCK. To produce the wave forms shown, the following sequence should be entered:

1. SYNC PROBE TO FREERUN CLOCK
2. OUTPUT PROBE PULSER HIGH/LOW/TOGGLE

Wave forms are shown, for all three OUTPUT PROBE PULSER commands, synchronized to the system freerun clock.

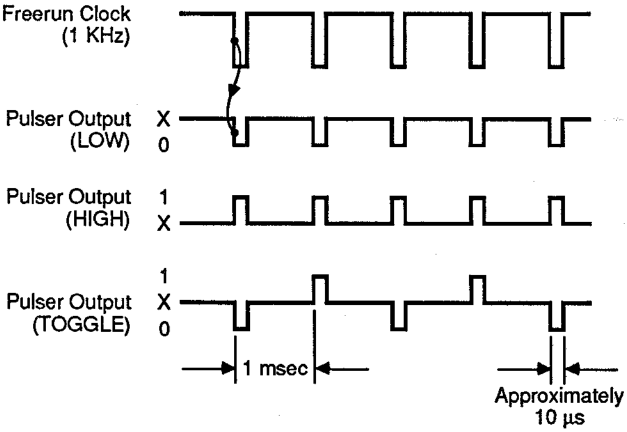


Figure 5-5: Freerun Sync (Probe) Example

Pod Sync (Probe) Example

Figure 5-6 is an example of the timing for the command SYNC PROBE TO POD ADDR. The example shows how the probe can be used both to output pulses and gather the resulting responses. To produce the wave forms shown, the following sequence should be entered:

1. SYNC PROBE TO POD ADDR
2. ARM PROBE FOR CAPTURE USING SYNC
3. OUTPUT PROBE PULSER HIGH/LOW/TOGGLE

Wave forms are shown for all three OUTPUT PROBE PULSER commands synchronized to the pod valid address cycle. In Figure 5-6, at points:

- **A:** ~PodSync falls; the probe output pulse starts.
- **B:** ~PodSync rises; the probe output pulse ends. At this edge, CRC signature and clocked level history are clocked at the probe input.
- **C:** To display responses clocked at all **B** since the ARM PROBE command, enter SHOW PROBE CAPTURED RESPONSES. The command also displays asynchronous responses gathered since ARM PROBE.

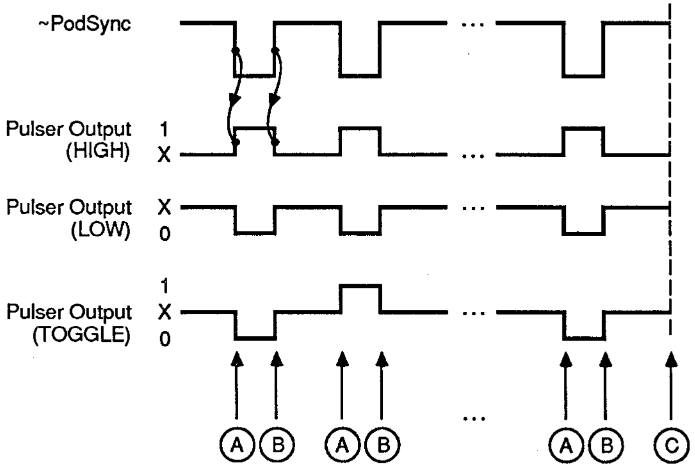


Figure 5-6: Pod Sync (Probe) Example

External Sync (Probe) Example 1

Figure 5-7 is an example of the timing for the command SYNC PROBE TO EXT ENABLE HIGH. The example shows how the probe can be used to output pulses and gather the resulting responses. To produce the wave forms shown, the following sequence should be entered:

1. SYNC PROBE TO EXT ENABLE HIGH CLOCK ↓ START ↑ STOP ↑
2. ARM PROBE FOR CAPTURE USING SYNC
3. OUTPUT PROBE PULSER HIGH/LOW/TOGGLE

Wave forms are shown for all three OUTPUT PROBE PULSER commands synchronized to the external clock module lines. In Figure 5-7, at points:

- **A:** Pulser output does not begin because the enabling condition (ENABLE HIGH) is not true at a falling clock edge.
- **B:** Output begins; the pulser turns on at the first enabled falling clock edge after Start.
- **C:** The pulser turns off at the next rising edge after it turns on. At this edge, CRC signature and clocked level history are clocked at the probe input.
- **D:** Stop, which does not have to be enabled, ends the synchronized period. To display responses clocked at all C since the ARM PROBE command, enter SHOW PROBE CAPTURED RESPONSES. The command also displays asynchronous responses gathered since ARM PROBE.

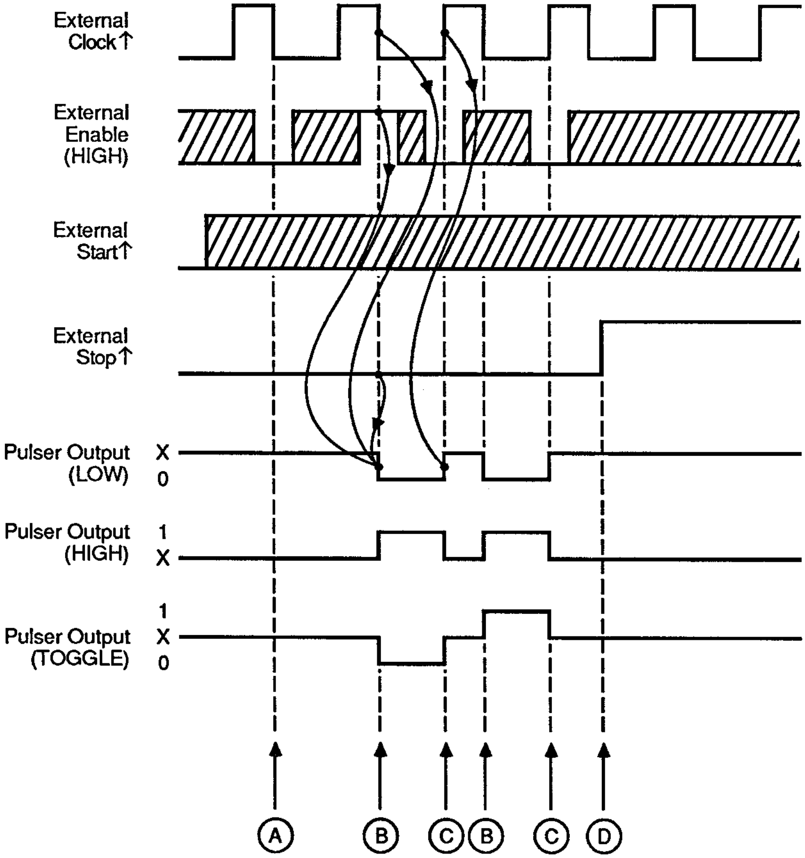


Figure 5-7: External Sync (Probe) Example 1

External Sync (Probe) Example 2

Figure 5-8 is an example of the timing for the command SYNC PROBE TO EXT ENABLE POD*EN0. The example shows how the probe can be used to output 4 pulses and gather the resulting responses. To produce the wave forms shown, the following sequence should be entered:

1. SYNC PROBE TO EXT ENABLE POD*EN0 CLOCK ↑ START ↑ COUNT 4
2. ARM PROBE FOR CAPTURE USING SYNC
3. OUTPUT PROBE PULSER LOW

Wave forms are shown for OUTPUT PROBE PULSER LOW synchronized to the external clock module lines. In Figure 5-8, at points:

- **A:** Start is true.
- **B:** The enabling condition (POD * ENO) becomes true.
- **C:** Output begins; pulser turns on at any falling clock edge after an enabled Start.
- **D:** Pulser turns on at all falling edges after **C**, as long as the enabling condition remains true.
- **E:** Pulser turns off at the next rising edge after it turns on. At this clock edge, CRC signature and clocked level history are clocked.
- **F:** Pulsing ends after 4 pulses, as specified by COUNT 4. To display responses clocked at all **E** since the ARM PROBE command, enter SHOW PROBE CAPTURED RESPONSES. The command also displays asynchronous responses gathered since ARM PROBE.

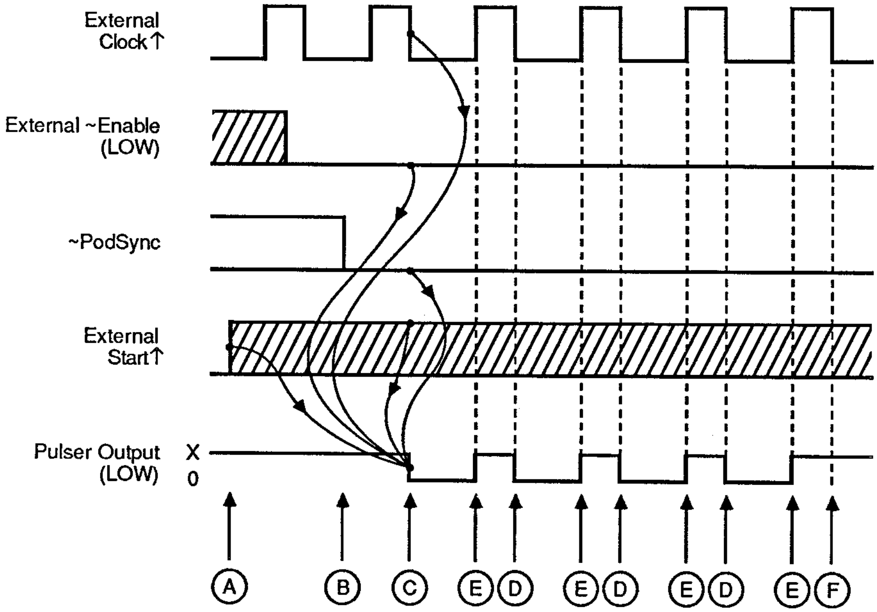


Figure 5-8: External Sync (Probe) Example 2

NOTE

The probe's pulser always turns off at the selected edge. Therefore, it turns on, if enabled, at the first available opposite edge to that selected. This is because data must be gathered at the trailing edge of the pulse.

As the preceding examples show, if the edge is defined as active when *rising*:

- Input occurs after enabling-condition-Start is true, at *rising* clock edges, and continues while the enabling condition is true.
- The pulser turns on at *falling* clock edges.
- The pulser turns off at the next *rising* clock edge after it turns on.

Pod Sync (Probe and I/O Module) Example

I/O module data gathering is clocked when the pod's PodSync line rises at the end of its synchronized period. Since the probe pulser turns on at the falling edge of PodSync, the probe can be used to output pulses while the I/O module measures the resulting responses.

Figure 5-9 is an example of the timing for the command SYNC I/O MOD TO POD ADDR. The example shows how the I/O module can be used to gather responses generated by probe output.

To capture data as shown, the following sequence should be entered:

1. SYNC I/O MOD TO POD ADDR.
   (This automatically enters the command:
   SYNC PROBE TO POD ADDR.)

2. ARM I/O MOD FOR CAPTURE USING SYNC

3. OUTPUT PROBE PULSER HIGH

Wave forms are shown for OUTPUT PROBE PULSER HIGH synchronized to the valid pod address cycle. In Figure 5-9, at points:

- **A:** ~PodSync falls low, turning on the probe output pulse.
- **B:** ~PodSync rises, ending the probe output pulse.
- **C:** The rising ~PodSync clocks I/O module gathering of CRC signatures and clocked level history.
- **D:** To end the synchronized period and display responses clocked at all **C** since the ARM I/O MOD command, enter SHOW I/O MOD CAPTURED RESPONSES. The command also displays asynchronous responses gathered since ARM I/O MOD.

External Sync (I/O Module) Example 1

Figure 5-10 is an example of the timing for the command SYNC I/O MOD TO EXT ENABLE. To capture data as shown, the

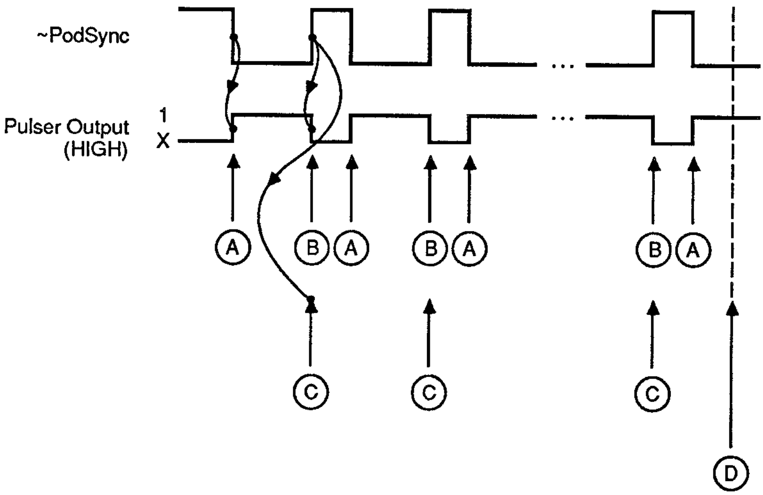


Figure 5-9: Pod Sync (Probe and I/O Module) Example

following sequence should be entered:

1. SYNC I/O MOD <n> TO EXT ENABLE HIGH CLOCK ↑ START ↑ STOP ↑

2. ARM I/O MOD FOR CAPTURE USING SYNC

Wave forms are shown for I/O module input, synchronized to external clock. In Figure 5-10, at points:

- **A:** Start edge is recognized but response gathering cannot begin until the enabling condition (ENABLE HIGH) becomes true.
- **B:** Active clock edge after Start is ignored because the enabling condition is not yet true.
- **C:** Enabling condition becomes true.
- **D:** The rising clock edge clocks gathering of CRC signatures and clocked level history.
- **E:** Enabling condition removed; asynchronous measurements stop.
- **F:** Stop ends response gathering. Stop does not have to be enabled.
- **G:** To display responses clocked at all **D** since the ARM I/O MOD command, enter SHOW I/O MOD CAPTURED RESPONSES. The command also displays asynchronous responses gathered since ARM I/O MOD.

In this example, there were three enabled clock edges which sampled high, high, and low levels respectively. There was only one enabled low-to-high transition. When SHOW I/O MOD is entered, the display should read:

```
SHOW I/O MOD <n> PIN <m> CAPTURED RESPONSES
SIG = 6        ASYNC LEVEL = 10
COUNT = 1      CLOCKED LEVEL = 10
```

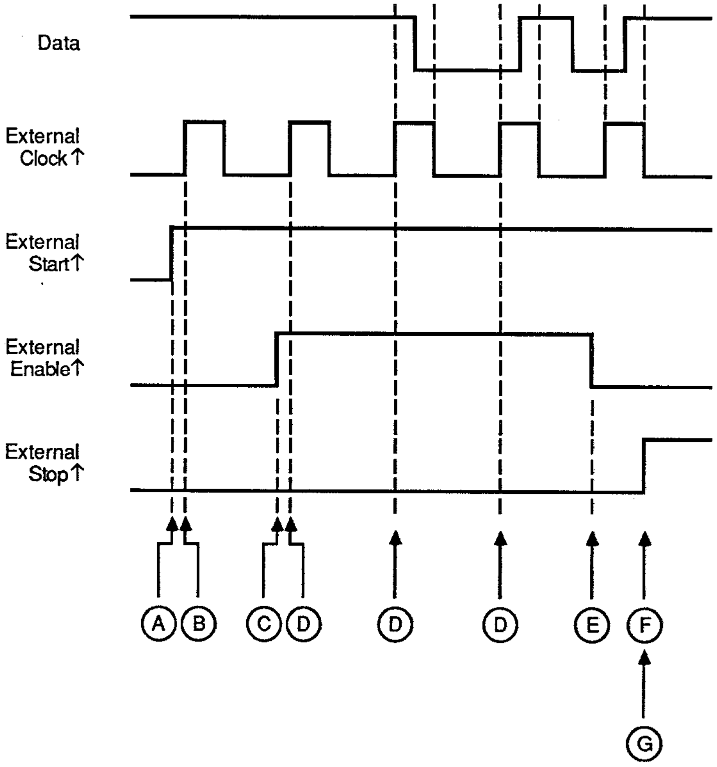


Figure 5-10: External Sync (I/O Module) Example 1

External Sync (I/O Module) Example 2

Figure 5-11 is an example of the timing for the command SYNC I/O MOD TO EXT ENABLE. To capture data as shown, the following sequence should be entered:

1. SYNC I/O MOD <n> TO EXT ENABLE POD CLOCK ↑ START ↑ STOP COUNT 3
2. ARM I/O MOD FOR CAPTURE USING SYNC

Wave forms are shown for I/O module input synchronized to external clock. In Figure 5-12, at points:

- **A:** Enabling condition (POD) becomes true.
- **B:** The active clock edge is ignored because no Start signal has been received.
- **C:** Start received. Since enabling condition is true, response gathering begins.
- **D:** Response gathering is clocked for 3 active clock edges as specified by COUNT 3.
- **E:** Stop signal is ignored because COUNT 3 was specified.
- **F:** Data gathering ends after 3 clock edges as specified by COUNT 3. To display responses clocked at all **D** since the ARM I/O MOD command, enter SHOW I/O MOD CAPTURED RESPONSES. The command also displays asynchronous responses gathered since ARM I/O MOD.

  In this example, all three enabled clock edges sampled high levels. There were three rising transitions between Start and Stop. When SHOW I/O MOD is entered, the display should read:

```
SHOW I/O MOD <n> PIN <m> CAPTURED RESPONSES
SIG = 7         ASYNC LEVEL = 10
COUNT = 3       CLOCKED LEVEL = 1
```

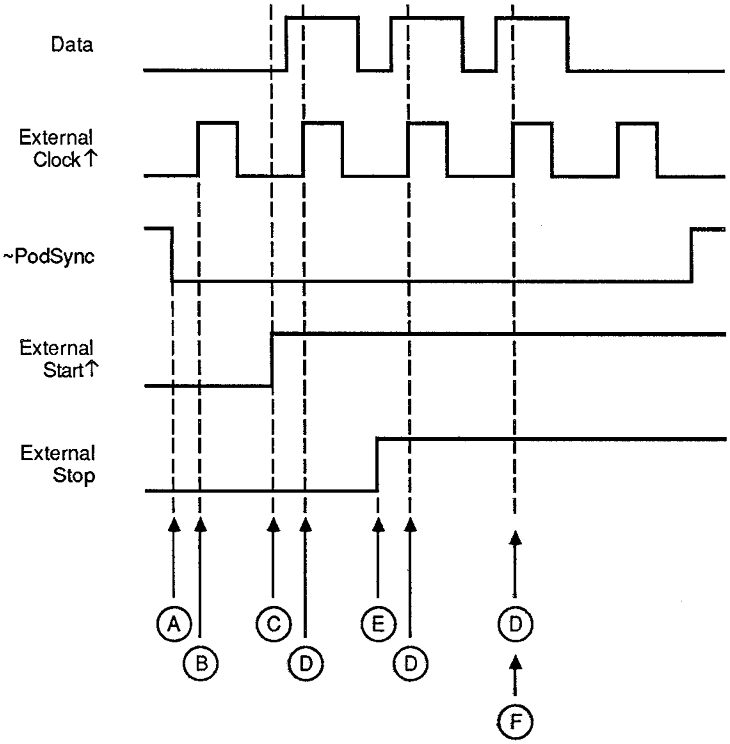
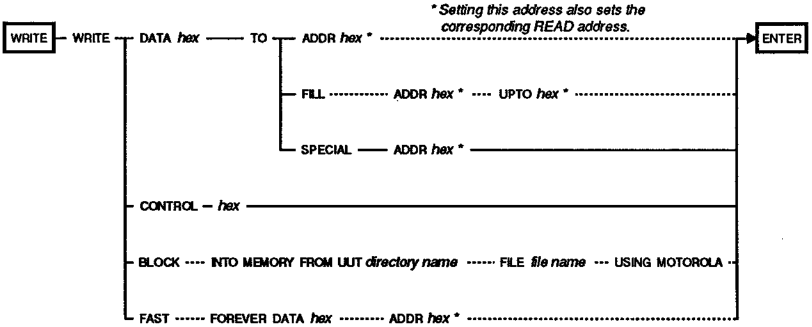


Figure 5-11: External Sync (I/O Module) Example 2

WRITE 5.30.

Writes data to specified memory, I/O, or other addresses. The contents of a file can be loaded into the pod RAM (for pods that have overlay RAM) or into the UUT.



Data can be written:

- To a UUT or pod memory address or range of addresses. This data could be loaded into the UUT or pod RAM from a file. The write cycle can be observed on an oscilloscope.
- To pod control lines to test whether they can be driven.

WRITE ...

DATA <hex> TO ...

Describes the data bit pattern to be written as a hex number.

ADDR <hex>

Writes the data to the specified address.

FILL ADDR <hex> UPTO <hex>

Writes the data to each address within the specified address range.

SPECIAL ADDR <hex>

Writes the data to the specified UUT virtual address or pod memory address. This command is pod dependent. It should only be used to access quick, pod-resident tests because it causes the system to lose some fault-handling capability.

The command can be used with older (9010-compatible) pods which use virtual addresses to distinguish between I/O and memory addresses, between bytes and words, between user memory and program memory, and so on. In the 9100A/9105A, the OPTION key is used for such purposes.

CONTROL <hex>

Momentarily drives pod control lines to test that they can be written to. Not all control lines can be driven in this way. Refer to the pod decal or manual to verify which ones can be driven.

WRITE ...

BLOCK INTO MEMORY FROM UUT <directory name> FILE <file name> USING MOTOROLA

Loads the contents of a file in the specified format into UUT or pod overlay RAM. The file must contain information about the starting address and number of data bytes.

You can use this command to load a program into UUT RAM before executing RUN UUT at a UUT RAM address.

FAST FOREVER ADDR <hex>

Command execution begins with one write operation at the specified address. This is followed by a very rapid sequence of writes at the same address. Connecting an oscilloscope's external trigger input to the 9100A/9105A TRIGGER OUTPUT triggers the oscilloscope at each write cycle.

Section 6
Test Techniques and Examples

INTRODUCTION 6.1.

This section illustrates the use of the 9100A/9105A in testing and troubleshooting using an 80286-based UUT and an 80286 pod as an example. Some of the examples are specific to the Demo/Trainer UUT, which is available as an option from Fluke. However, where possible, the examples are general. Signatures and other responses indicated in the examples are those actually measured on the Demo/Trainer UUT with the 9100A or 9105A using the current software and hardware available at the time of publication.

NOTE

Each Demo/Trainer UUT example should begin with the UUT set to the "no-fault" state. In this state, its eight SW1 switches are CLOSED and all the other fault switches (on SW2 through SW6) are OPEN.

It is assumed that you know how to use the keypad and are acquainted with the commands used in the following examples. Section 5, "Keypad Reference," provides detailed descriptions of all of the commands.

BUILT-IN TESTS AND STIMULI 6.2.

You can use the operator's keypad to execute functions built into the 9100A/9105A, which perform functional tests of the UUT. The following built-in functions are rapid tests which provide very comprehensive fault information:

- BUS: tests the UUT microprocessor bus.
- RAM: performs FAST (probabilistic) or FULL (deterministic) test of the UUT RAM.
- ROM: test the UUT ROM by comparing its signature to that from a known-good UUT ROM.

This section illustrates the use of the above tests and also shows how the built-in stimuli can be used to test a UUT. The stimuli (READ, WRITE, and STIM) are effected by the pod, and the results are monitored by the probe or I/O module.

BUS Test 6.2.1.

This test checks the UUT microprocessor bus for lines that are stuck high or low, and lines that are tied together. To start the test, use the BUS key to enter the command:

```
TEST BUS AT ADDR 0
```

If the UUT has faults, the STOPPED light should come on to show that the system stopped at the first fault found. The user can then LOOP on the fault or press CONT to let the BUS test continue. In the latter case, if no more faults are encountered, the test will finish, after which the BUSY light goes off.

The system reports faults using the microprocessor-specific information in the pod database. The faults may include pod-specific names for signals as well as pin numbers.

Address Line Tied Low

1. Tie A4 at the UUT microprocessor to common. On the Demo/Trainer UUT, close SW2-5.

2. Start the BUS test.

3. The STOPPED light comes on. The display describing the fault should be:

   ```
   TEST BUS AT ADDR 0
   addr line A4 pod pin 27 stuck low
   ```

4. Press CONT. The test will now finish.

Two Address Lines Tied Together

1. Tie A8 and A5 together at the UUT microprocessor. On the Demo/Trainer UUT, close SW2-4.

2. Start the BUS test.

3. The STOPPED light comes on. The display describing the fault should be:

   ```
   TEST BUS AT ADDR 0
   addr line A8 pod pin 23 tied
   ```

4. Press CONT. The STOPPED light comes on and a second fault message is displayed.

   ```
   TEST BUS AT ADDR 0
   addr line A5 pod pin 26 tied
   ```

5. Press CONT. The test will now finish.

Address Line Tied High

1. Tie A9 at the UUT microprocessor to logic high. On the Demo/Trainer UUT, close SW2-3.

2. Start the BUS test.

3. The STOPPED light comes on. The display describing the fault should be:

```
TEST BUS AT ADDR 0
addr line A9 pod pin 22 stuck high
```

4. Press CONT. The display should then be:

```
TEST BUS AT ADDR 0
addr line A9 pin 22 not drivable
```

5. Press CONT. The test will now finish.

Two Data Lines Tied Together

1. Tie D9 and D8 together at the UUT microprocessor. On the Demo/Trainer UUT close SW2-8.

2. Start the BUS test.

3. The STOPPED light comes on. The display describing the fault should be:

```
TEST BUS AT ADDR 0
data line D8 pod pin 37 tied
data line D9 pod pin  39 tied
```

4. Press CONT. The test will now finish.

Interrupt Active

1. Tie the interrupt line at the UUT microprocessor to logic high. On the Demo/Trainer UUT, close SW3-2.

2. Enable interrupts by entering:

```
SETUP POD REPORT INTR ACTIVE ON
```

3. Start the BUS test.

4. The STOPPED light comes on. The display describing the fault should be:

```
TEST BUS AT ADDR 0
interrupt INTR pod pin 57 active
```

If POD REPORT INTR ACTIVE had been set to OFF in step 2, the BUS test would not report this fault.

RAM Test 6.2.2.

Open Address Bus Line

1. Open address line A15 between the microprocessor socket and the rest of the UUT. This can be done by opening pin 16, a header between the pod and the UUT microprocessor socket. On the Demo/Trainer UUT, open SW1-1.

2. Start the BUS test. No faults will be detected; however, this type of fault can be found by the RAM test.

3. Start TEST RAM FAST over the UUT's memory address range (0 through 1FFFE in the Demo/Trainer UUT). The default ADDR STEP (2), DELAY (250), and SEED (0) should be used. Set OPTION to the UUT addressing mode (MEMORY, BYTE or WORD).

4. The STOPPED light comes on. The display describing the fault should be:

```
TEST RAM FAST ADDR 0 UPTO 1FFFE DATA ...
... MASK FFFF ADDR STEP 2 DELAY 250 SEED 0
attempted to write at 0
addr line A15 stuck or open
```

5. The probe is to be used for the next stage. Enter:

```
SYNC PROBE TO POD ADDR
```

   or enter:

```
SYNC PROBE TO POD DATA
```

6. The STIM command can now be entered and used to locate the open circuit in conjunction with the probe. For example, you could enter:

```
TOGGLE ADDR 0 MASKED BY 8000
```

7. While the STIM command of step 6 exercises A15, probe A15 after entering the command:

```
INPUT PROBE LEVEL
```

   A15 is traced from the microprocessor socket out through the bus buffers until an expected response is not received at a particular node. You should test all pins forming a node when looking for open circuits. (More on this topic can be found in the *Applications Manual.*)

Dynamic RAM Row Address Fault

1. Open the UUT row address strobe (RAS) line to a bank of dynamic memory. On the Demo/Trainer UUT, open SW1-4.

   Opening the RAS line means that the memory cannot be selected; all writes and reads to dynamic RAM are non-functional.

2. Start TEST RAM FAST over the UUT's memory address range (0 through 1FFFE in the Demo/Trainer UUT). The default ADDR STEP (2), DELAY (250), and SEED (0) should be used. Set OPTION to MEMORY WORD. The addresses must be even for a 16-bit bus and word addressing.

3. The STOPPED light comes on. The display describing the fault should be:

   ```
   TEST RAM FAST ADDR 0 UPTO 1FFFE DATA ...
   ... MASK FFFF ADDR STEP 2 DELAY 250 SEED 0
   attempted to write at 0 read FFFF
   cannot modify RAM data
   ```

   From the fault message reported (all data lines stuck high), we can deduce that the dynamic RAM is not being selected.

Dynamic RAM Address Multiplexer Fault

1. Short the UUT dynamic RAM address multiplexer output A5/A13 high. On the Demo/Trainer UUT, close SW4-8.

2. Start TEST RAM FAST over the UUT's memory address range (0 through 1FFFE in the Demo/Trainer

UUT). The default ADDR STEP (2), DELAY (250), and SEED (0) should be used. Set OPTION to MEMORY WORD. The addresses must be even for a 16-bit bus and word addressing.

3. The STOPPED light comes on. The display, describing the fault, should be:

```
TEST RAM FAST ADDR 0 UPTO 1FFFE DATA ...
... MASK FFFF ADDR STEP 2 DELAY 250 SEED 0
attempted to read at 0
addr line A5 stuck or open
addr line A13 stuck or open
```

The system finds the fault indirectly, by reporting that address lines A5 and A13 are not responding. In effect, it cannot tell the difference between the lines. The problem would be carried over to any circuits common to A5 and A13.

Dynamic RAM Column Address Decoder Fault

1. Short the UUT's lower bank (D0-D7) dynamic RAM column address strobe (CAS) line to logic low. On the Demo/Trainer UUT, close SW5-5.

   This means that the lower bank cannot be selected because writes and reads to it are non-functional.

2. Start TEST RAM FAST over the UUT's memory address range (0 through 1FFFE in the Demo/Trainer UUT). The default ADDR STEP (2), DELAY (250), and SEED (0) should be used. Set OPTION to MEMORY WORD. The addresses must be even for a 16-bit bus and word addressing.

Run the test first with the data mask set to FF and then again using FF00 as the data mask.

3. With the mask set to FF, the STOPPED light comes on. The display describing the fault should be something like the following (depending on what data is latched when the fault switch SW1-5 is opened):

```
TEST RAM FAST ADDR 0 UPTO 1FFFE DATA ...
... MASK FF ADDR STEP 2 DELAY 250 SEED 0
attempted to write at 0 read E0
cannot modify RAM data
```

4. With the mask set to FF00, the test will pass and no fault message will appear on the operator's display.

Open Data Line Fault

1. Open data line D8 output of the corresponding memory chip. This can be done by lifting the pin from the socket. On the Demo/Trainer UUT, open SW1-5.

2. Start TEST RAM FAST over the UUT's memory address range (0 through 1FFFE in the Demo/Trainer UUT). The default ADDR STEP (2), DELAY (250), and SEED (0) should be used. Set OPTION to MEMORY WORD. The addresses must be even for a 16-bit bus and word addressing.

3. The STOPPED light comes on. The display describing the fault should be:

```
TEST RAM FAST ADDR 0 UPTO 1FFFE DATA ...
... MASK FFFF ADDR STEP 2 DELAY 250 SEED 0
attempted to write data 0 at 0 read 100
data line D8 stuck high
```

The system reports data line D8 stuck high because it floats high when it is open, and the system cannot distinguish this case from an open line. The open circuit could be detected using the synchronized probe and the STIM command stimuli.

Dynamic RAM Refresh Dead Fault

1. Disable the refresh of your dynamic memory system. On the Demo/Trainer UUT, close SW5-1.

2. Start TEST RAM FAST over the UUT's memory address range (0 through 1FFFE in the Demo/Trainer UUT). The DELAY should be set to 60000 msec. The default ADDR STEP (2), and SEED (0) should be retained.

NOTE

> *Most dynamic RAM must be refreshed every 2 msec. However, it has been found that dynamic RAM may retain its contents for almost a minute after refresh is removed. Therefore, to observe a refresh fault, a long delay of 60000 msec should be used.*

3. The STOPPED light comes on. The display describing the fault will depend on when the refresh failed, but should be something like:

```
TEST RAM FAST ADDR 0 UPTO 1FFFE DATA ...
... MASK FFFF ADDR STEP 2 DELAY 60000 ...
... SEED 0
attempted to write at FE
read incorrect data A876 expected A976
```

The values of data read and written and the address are UUT dependent. The important point is that the time between each memory access is lengthened to allow refresh problems to surface.

Storing ROM Signatures

Before testing the ROM of a suspected bad UUT, ROM signatures from a known-good UUT must be stored. It is recommended that a signature be stored under the corresponding REF (reference designator) for each ROM chip. These signatures can be stored on disk using the SETUP SAVE SYSTEM SETTINGS command.

Assume that your UUT has four ROM chips, two each for the data-bus high and low bytes. A signature must be stored for each chip, for that chip's address range. In the following Demo/Trainer UUT examples substitute your UUT ROM chip's address range (ADDR ... UPTO) and reference designator.

To get a signature from the low-byte ROM on a known-good UUT:

1. Enter the command:

```
GET SIG ROM REF U27 ADDR F0000 UPTO ...
... FFFFE MASK FF ADDR STEP 2
ADDR OPTION:  MEMORY BYTE
```

   The signature is displayed as "SIG = F387".

2. Enter the command:

```
GET SIG ROM REF U29 ADDR E0000 UPTO ...
... EFFFE MASK FF ADDR STEP 2
ADDR OPTION:  MEMORY BYTE
```

   The signature is displayed as "SIG = 8E6E".

To get a signature from the high-byte ROM on a known-good UUT:

1. Enter the command:

   ```
   GET SIG ROM REF U28 ADDR F0001 UPTO ...
   ... FFFEF MASK FF ADDR STEP 2
   ADDR OPTION:  MEMORY BYTE
   ```

   The signature is displayed as "SIG = 8E6E".

2. Enter the command:

   ```
   GET SIG ROM REF U30 ADDR E0001 UPTO ...
   ... EFFFF MASK FF ADDR STEP 2
   ADDR OPTION:  MEMORY BYTE
   ```

   The signature is displayed as "SIG = F387".

The signature could also be gathered using the MEMORY WORD address option, with an appropriate mask for the tested chip. Using this method for U28, the high-byte ROM, enter the following command and address option:

```
GET SIG ROM REF U28 ADDR F0000 UPTO ...
... FFFFE MASK FF00 ADDR STEP 2
ADDR OPTION:  MEMORY WORD
```

The signature is displayed as "SIG = 8E6E".

ROM Signature Fault (Example 1)

1. Open the output enable of a UUT ROM chip. On the Demo/Trainer UUT, open SW1-2.

2. Start TEST ROM FULL for the component with the bad output enable. On the Demo/Trainer UUT, it is U27.

3. The STOPPED light comes on. The display describing the fault should be:

```
TEST ROM FULL REF U27
testing from addr F0000 to FFFFE
all ROM data bits stuck high
```

The data returned is all "1's" because the chip cannot be selected.

ROM Signature Fault (Example 2)

1. Tie the chip select of a UUT ROM chip low, forcing it to be enabled always. On the Demo/Trainer UUT, close SW4-3.

   A ROM test of this chip would not indicate a fault. However, testing other chips that share the same data lines will produce an incorrect ROM signature. These faults arise from the chip that tested good by itself. The faults are caused from data bus contention between the ROMs.

2. Start TEST ROM FULL for the component with the bad chip enable. On the Demo/Trainer UUT, it is U27.

3. The system will not have found a fault; the BUSY light will go off.

4. Start TEST ROM FULL for a component that shares data lines with the ROM chip in step 2. On the Demo/Trainer UUT, the component is U29.

5. The STOPPED light comes on. The display describing the fault should be:

```
TEST ROM FULL REF U29
testing from addr E0000 to EFFFE
read incorrect sig xxxx expected 8E6E
```

"xxxx" is a non-stable signature which varies from UUT to UUT. From this example it can be seen that the ROM chip that fails is not always the chip that has the fault. For troubleshooting this type of failure see the *Applications Manual.*

Single Bit ROM Signature Fault

1. Tie data line D5 of one of the UUT ROM chips low. On the Demo/Trainer UUT, close SW4-2.
2. Start TEST ROM FULL for any of the ROM chips. On the Demo/Trainer UUT, the chip should be U27.
3. The STOPPED light comes on. The display describing the fault should be:

```
TEST ROM FULL REF U27
testing from addr F0000 to FFFFE
data line D5 stuck low
```

TESTING USING THE PROBE AND THE I/O MODULE 6.3.

Introduction

The probe and I/O module are used in conjunction with the pod to test or troubleshoot UUT components that can be stimulated by the pod, but cannot be read using the pod. Similar testing or troubleshooting can also be done on components that cannot be stimulated by the pod. This is done by using the drive features of the probe and I/O module. The probe and the I/O module have very similar capabilities, which are listed and compared on the following two pages.

	Probe	*I/O Module*
Channels	1	40 per module
Data Thresholds	TTL, CMOS, RS232	TTL, CMOS
Start, Stop, Enable, and Clock thresholds	TTL	TTL
Synchronous level history	1, X, or 0. Externally or pod clocked.	1, X, or 0. Externally or pod clocked.
Asynchronous level history	1, X, or 0. Non-clocked.	1, X, or 0. Non-clocked.
Snapshot logic level	Level of probe at at command entry.	Level of I/O pins at command entry.
Logic-level lights	1 (red), X (yellow), 0 (green) indicators.	None
Transition counts	24 bits plus overflow with external start, stop, and enable.	23 bits plus overflow with external start, stop, and enable.
Frequency	40 MHz max	10 MHz max

(continued on next page)

(continued from previous page)

	Probe	*I/O Module*
CRC signature	16-bit pod clocked or externally clocked.	16-bit pod clocked or externally clocked.
Clock	Freerun, PodSync, or external, rising or falling edge. Also internal.	PodSync or external, rising or falling edge. Also internal.
Enable*	Always, High, Low, or during PodSync. Also PodSync and Enable Low, or PodSync and Enable High.	Always, High, Low, or during PodSync.
Start*	Rising or falling edge.	Rising or falling edge
Stop*	Rising or falling edge, or after 1 to 65,535 clocks.	Rising or falling edge, or after 1 to 65,535 clocks.
Output	Output pulsed high or low, or toggled high/low. Sync to external, free-run clock, or to PodSync.	Output latched or pulsed, high or low, on any of 40 pins.

* Only used with the external CLOCK line.

NOTE

I/O modules are connected to DIP components by clip modules. The part number of a clip module tells you the size of the DIP it clips over. For example, a 28-pin clip module is labeled "Y9100A-28D."

CRC Signatures 6.3.2.

Both the probe and the I/O module can characterize a node by taking a CRC signature. The clock used to gather the CRC can be either an external clock or PodSync. When using the external clock, the data gathering window is controlled by the START, STOP, and ENABLE lines as explained in the SYNC section of Section 5, "Keypad Reference."

Probe (Externally Synchronized)

This example illustrates how the probe can take signatures synchronized to external events using the clock module. The circuit tested is the refresh control of the Demo/Trainer UUT dynamic RAM . This technique could be applied to almost any dynamic RAM refresh circuit.

Some UUTs require setup of control circuits to enable refresh circuitry. This is accomplished by writing out the appropriate data using the WRITE command. Initialization could be done with individual writes or stored and executed using the WRITE BLOCK command. It could also be done with a TL/1 program. The Demo/Trainer UUT requires no initialization for memory refresh.

1. Figure 6-1 shows how the Start, Stop, Enable, and Clock signals should be chosen for the test

   Start: Rising edge of refresh request (RFRQ).

Stop: A stop signal is not used. Instead, the sync period ends after 1000 clock pulses allowing multiple refresh cycles to be observed. If the user used the end of RFRQ the user would only see one refresh cycle.

Enable: Enable low during refresh grant (~RFGT).

Clock: Clock on the rising edge of the refresh controller state machine's clock.

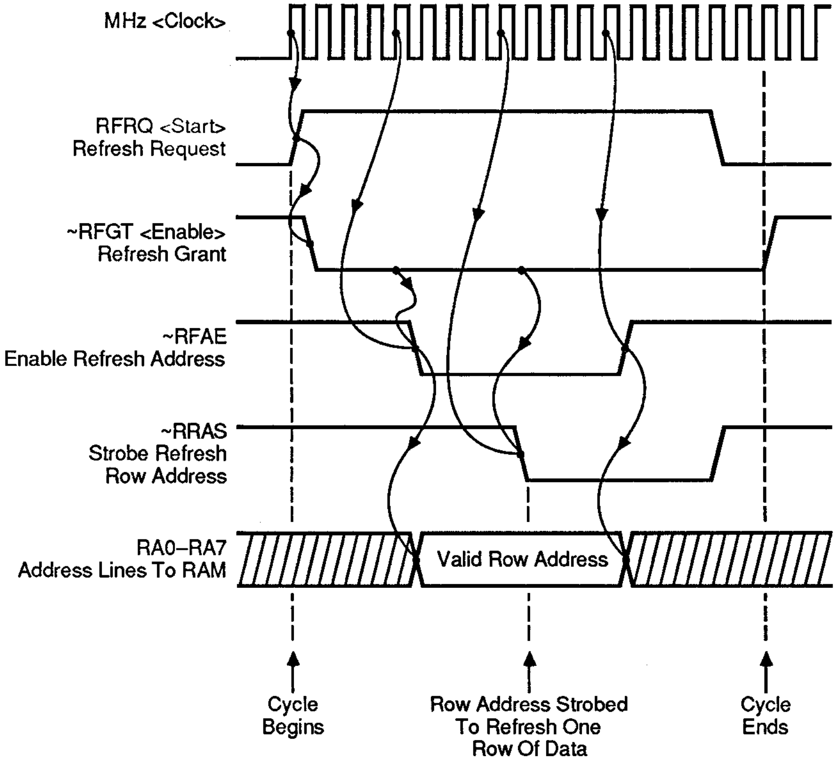
2. Connect the clock module external sync lines to the following UUT pins:

```
START (green)   - U43-11
CLOCK (yellow)  - U25-1
ENABLE (blue)   - U61-11
COMMON (black)  - any convenient UUT common
```

3. Enter the command:

```
SYNC PROBE TO EXT ENABLE LOW CLOCK ↑ ...
... START ↑ STOP COUNT 1000
```

4. Take and display each signature using an ARM PROBE command followed by a SHOW PROBE command.



Stop After a Count of 1000 Clock Pulses
<Clock>, <Start>, & <Enable>

Figure 6-1: Synchronizing to Dynamic RAM Refresh Cycle

The following is a listing of the correct signatures:

Probe		*Signature*
U43-11	RFRQ	1C8B
U61-11	~RFGT	0
U59-10		1C8B
U59-9	~RFGT	0
U64-13		C4F
U44-5	RFAE	4313
U44-6	~RFAE	5F98
U44-9	RRAS	10C4
U44-8	~RRAS	C4F
U64-10		0
U63-8	~RASS	C4F
U58-8	~CASU	1C8B
U58-11	~CASL	1C8B
U26-8	~RAM_WRITE	1C8B

NOTE:

In the following examples, to avoid mistakes when taking signatures using PodSync, always follow this sequence:

1. *Using the SYNC key, synchronize the probe to the period when the data being measured is true.*

2. *Use the ARM (PROBE or I/O MOD) command to initialize the signature-gathering hardware.*

3. *Apply a stimulus to the node under test using the STIM key while taking data from the node.*

4. *Use the SHOW (PROBE or I/O MOD) command to display the results of the measurement.*

Probe (Pod Synchronized)

Here is an example of how to gather a synchronized signature at data bit D0:

1. Synchronize the probe to pod data using:

```
SYNC PROBE TO POD DATA.
```

2. Connect the probe to data line D0.

3. Arm the probe:

```
ARM PROBE FOR CAPTURE USING SYNC
```

4. Use the RAMP command as a stimulus to the lower byte of the data bus.

```
RAMP DATA 00 MASKED BY FF AT ADDR 0
```

5. Display the signature using the SHOW function.

```
SHOW PROBE CAPTURED RESPONSES
```

6. The displayed results should be:

```
SIG= 96EC   ASYNC LEVEL = 1 0
COUNT = 128   CLOCKED LEVEL = 1 0
```

I/O Module (Pod Synchronized)

To gather pod-synchronized signatures with the I/O module from the low-order byte of a UUT data bus buffer:

1. Attach the 20-pin clip module to the "A" side of the I/O module. Connect the clip to the data bus low-byte buffer. On the Demo/Trainer UUT, clip U3.

2. Synchronize the I/O module to the pod's valid data cycle:

   ```
   SYNC I/O MOD 1 TO POD DATA
   ```

3. Arm the I/O module:

   ```
   ARM I/O MOD 1 FOR CAPTURE USING SYNC
   ```

4. Using the RAMP command, ramp the lower 8 bits of the data bus.

   ```
   RAMP DATA 00 MASKED BY FF AT ADDR 0
   ```

5. To display the data, enter the command:

   ```
   SHOW I/O MOD 1 PIN 11 CAPTURED RESPONSES
   ```

6. The displayed results should read:

   ```
   SIG = 96EC    ASYNC LEVEL = 1 0
   COUNT = 128    CLOCKED LEVEL = 1 0
   ```

   The advantage of using the I/O module over the probe is the ability to simultaneously measure responses from several pins. To observe the responses at another data line, simply repeat step 5 for the required pin.

Transition Counting 6.3.3.

Transition counting is used to characterize a node that will not yield a stable CRC. It is not as definitive as a CRC signature, but it is better than a frequency or level measurement.

The probe and I/O module counters can be controlled with external START, STOP, and ENABLE lines. In addition both counters can display the number of counts between a manual start and stop (entered from the keypad). Counts can be gathered on all I/O modules simultaneously.

Probe (Externally Enabled)

1. Choose the Start, Stop, Enable and Clock signals for the test as follows:

   Start: Rising edge of refresh request (RFRQ).

   Stop: A stop signal is not used. Instead, the sync period ends after 1000 clock pulses allowing multiple refresh cycles to be observed. If the end of RFRQ were specified, you would see only one refresh cycle.

   Enable: Enable low during refresh grant (~RFGT).

   Clock: Clock on the falling edge of the system clock.

2. Connect the clock module measurement control lines to the following UUT pins:

   ```
   START (green)   - U43-11
   CLOCK (yellow)  - U25-1
   ENABLE (blue)   - U61-11
   COMMON (black)  - any convenient UUT common
   ```

3. Enter the command:

```
SYNC PROBE TO EXT ENABLE LOW CLOCK ↓ ...
... START ↑ STOP COUNT 1000
```

4. Connect the probe to U56-12 of the Demo/Trainer UUT.

5. Arm the probe for capture:

```
ARM PROBE FOR CAPTURE USING SYNC
```

6. Display the probe responses:

```
SHOW PROBE CAPTURED RESPONSES
```

7. The results should be:

```
COUNT = 125
```

The count value is the only significant response.

I/O Module (Pod Enabled)

1. Disable the dynamic memory refresh request by connecting the clear line of U43 to low. On the Demo/Trainer UUT, close SW5-1.

2. Choose the Start, Stop, and Clock signals for the test:

Start: Falling edge of the RAM decoder enable.

Stop: A stop signal is not used. Instead, the sync period ends after 24 clock pulses.

Clock: Clock on the rising edge of CLK, the RAM timing state machine's clock.

3. Connect I/O module mesurement control lines to the following UUT pins:

```
START (green)   - U7-5
CLOCK (yellow) - U13-1
COMMON (black) - any convenient UUT common
```

4. In the external sync mode, the pod enabling condition must also be set. Synchronize the I/O module to any pod bus cycle (Suppose you choose BUSCYCL for the 80286 pod):

```
SYNC I/O MOD 1 TO POD BUSCYCL
```

5. Enter the command:

```
SYNC I/O MOD 1 TO EXT ENABLE POD ...
... CLOCK ↓ START ↓ STOP COUNT 24
```

6. Attach a 20-pin clip module to the "A" side of the I/O module. Clip U15 on the Demo/Trainer UUT.

7. Arm the I/O module as follows:

```
ARM I/O MOD 1 FOR CAPTURE USING SYNC
```

8. Generate stimuli by performing the following reads and writes:

```
WRITE DATA 5555 TO ADDR 0
READ ADDR 0

WRITE DATA AAAA TO ADDR 0
READ ADDR 0

WRITE DATA 1234 TO ADDR 0
READ ADDR 0
```

9. To display transition counts, enter the command:

```
SHOW I/O MOD 1 PIN <x> CAPTURED RESPONSES
```

For "x", type the pin you desire to see. The transition counts should be:

I/O Module Pin	*IC Pin*	*Count*
3	3	3
5	5	6
16	16	6
19	19	3

Frequency Measurements 6.3.4.

A frequency measurement is used to characterize a node that has a repetitive waveform and cannot be characterized using a CRC signature or a transition count. Clocks are typically characterized by a frequency measurement.

Probe Frequency Counting

On the Demo/Trainer UUT, we could test that the clock divider, U25, and the clock, U1, are working properly by measuring frequencies using the probe.

1. To measure frequency with the probe, enter the command:

   ```
   FREQ AT PROBE
   ```

   Each time you press ENTER, the system will take a measurement and display the frequency in Hz.

2. Repeat step 1 while probing each of the following points; the frequencies close to those listed below should be displayed:

Probe	*Frequency*
U25-13	31939900 Hz
U25-9	15969950 Hz
U25-5	7984975 Hz
U1-10	7984975 Hz
U1-13	3992487 Hz

The 5 least-significant digits of the reported frequency may vary from UUT to UUT.

I/O Module Frequency Counting

The I/O module, like the probe, can be used to measure frequencies. The I/O module can do so, simultaneously, on several pins of a UUT component.

This example tests U43, a divide-by-16 IC on the Demo/Trainer UUT. We will use a 14-pin clip module. The I/O module pin mapping to a 14-pin clip module is as follows:

Clip Module Pin (IC)	*I/O Module Pin*
1-7	1-7 respectively
8-14	14-20 respectively

1. Attach a 14-pin clip module to side "A" of the I/O module. Clip U43 on the Demo/Trainer UUT.

2. To measure the frequency at U43-1, enter the command:

```
FREQ ON I/O MOD 1 PIN 1
```

3. Press ENTER to make a measurement.

4. Repeat step 3 for all pins (1 through 14) on U43. You can use the INC or DEC softkey to change the I/O module pin number appropriately before you press ENTER. When you press ENTER, the frequency at the tested pin should be displayed. Results for all U43 are summarized below. The reported frequencies may vary by ±2 Hz.

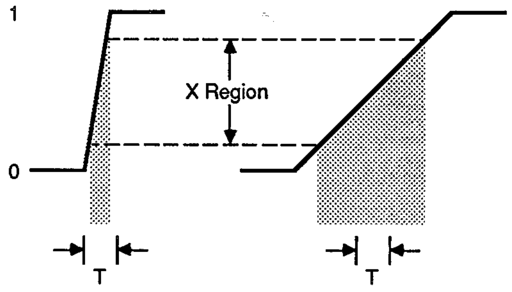
I/O Module Pin	*IC Pin*	*Frequency*
1	U43-1	0 Hz
2	U43-2	0 Hz
3	U43-3	66660 Hz
4	U43-4	66660 Hz
5	U43-5	66660 Hz
6	U43-6	66660 Hz
7	U43-7	0 Hz
14	U43-8	399952 Hz
15	U43-9	66660 Hz
16	U43-10	66660 Hz
17	U43-11	66660 Hz
18	U43-12	0 Hz
19	U43-13	0 Hz
20	U43-14	0 Hz

Logic Level Measurements 6.3.5.

Logic level measurements are used to trace a particular signal through a circuit. The 9100A/9105A can synchronize this measurement to pod microprocessor cycles or to external events. Level measurements can also be used to characterize a node if CRC signature, transition count, or frequency measurement do not give a stable measurement.

The probe and I/O module can sense 1, 0, and X (invalid) levels when measuring current level or when recording a synchronous or an asynchronous level history. When the probe or I/O module is in any sync mode *except* freerun sync, "X"-level sensing is filtered. Filtering applies to the recorded asynchronous level history and to the probe's yellow light. The feature is designed to avoid reporting an "X" during rapid 1-to-0 or 0-to-1 transitions, which briefly pass through the X region.

In the left-hand figure below, the signal changes rapidly from 0 to 1, spending less than a pre-defined time "T" in the X region. This X level would not be reported in the asynchronous level history.

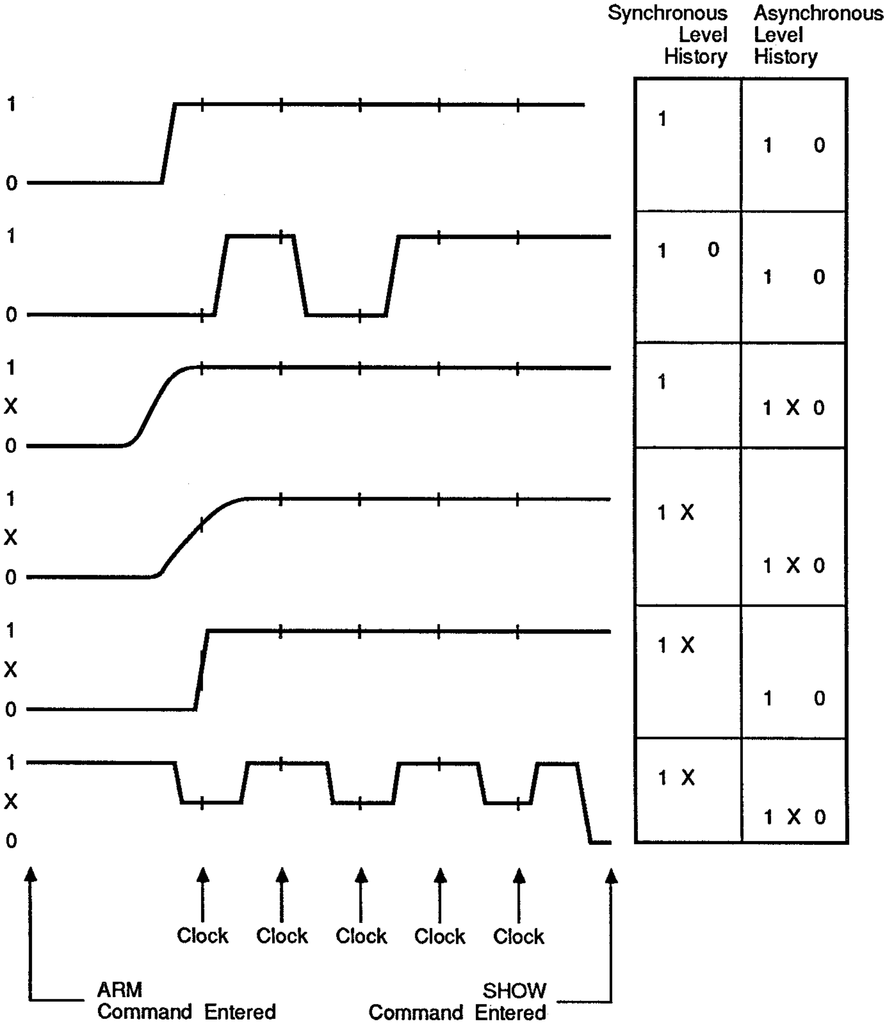


In the right-hand figure above, the transition is slow, spending more than time "T" in the X region. This X level would be reported in the asynchronous level history.

Values for "T" are shown below:

Device	*T*
Probe (TTL'or CMOS)	100 nsec
Probe (RS-232)	2 µsec
I/O module	100 nsec

Figure 6-2 shows a number of X-sensing examples recorded in the asynchronous and synchronous (clocked) level histories for different signals. For all the examples, the five indicated clock edges occur in the sync period between ARM (I/O MOD or PROBE) and SHOW (I/O MOD or PROBE). Slow and fast transitions are indicated by the previous convention.



Synchronous Level History	Asynchronous Level History
1	1 0
1 0	1 0
1	1 X 0
1 X	1 X 0
1 X	1 0
1 X	1 X 0

Figure 6-2: Level History Examples

Probe Level (Snapshot)

When the INPUT PROBE LEVEL command is entered, two pieces of information are displayed:

- Level at command entry.
- Asynchronous level since the last INPUT command was entered.

To measure logic levels with the probe:

1. Enter the command:

   ```
   SYNC PROBE TO FREERUN CLOCK
   ```

2. Enter the INPUT PROBE LEVEL command. The current level and asynchronous level history should be:

   ```
   INPUT PROBE LEVEL = X
       LEVEL HISTORY = X
   ```

   The results indicate that the current level on the probe is in the invalid region, and since the last input statement, the probe has only seen an invalid condition at the input.

3. Place the probe on the UUT common and do not remove it until step 6.

4. Press ENTER to repeat the INPUT PROBE LEVEL command. The display should read:

   ```
   INPUT PROBE LEVEL = 0
       LEVEL HISTORY =  X0
   ```

5. With the probe still in place, press the ENTER key. The display should read:

```
INPUT PROBE LEVEL = 0
    LEVEL HISTORY = 0
```

Since the probe was never removed from UUT common, the level history is now only 0 (low).

6. Place the probe on the UUT +5 V power source and press the ENTER key. The display should read:

```
INPUT PROBE LEVEL = 1
    LEVEL HISTORY = 1X0
```

The first line shows the current level (high). The second line shows the asynchronous level history since the last time the INPUT PROBE LEVEL command was entered in step 5. The history reflects:

- The current level (high).
- The floating level caused by moving the probe between common (step 5) and +5 V (step 6).
- The previous level (low) from step 5.

Probe Level (Pod Synchronized)

This example illustrates the use of the ARM PROBE and SHOW PROBE functions in observing asynchronous level and clocked level information.

1. Synchronize the probe:

```
SYNC PROBE TO POD ADDR
```

2. Arm the probe for capture:

```
ARM PROBE FOR CAPTURE USING SYNC
```

3. Enter SHOW PROBE, to see the results of previous captured response. The display should read:

   ```
   SHOW PROBE CAPTURED RESPONSES
   SIG= 0    ASYNC LEVEL = X
   COUNT = 0    CLOCKED LEVEL =
   ```

   ASYNC LEVEL captures level information independently of the clock. There were no clocked levels at the probe.

4. Place the probe on address line A2.

5. Arm the probe:

   ```
   ARM PROBE FOR CAPTURE USING SYNC
   ```

6. Generate the stimulus:

   ```
   WRITE DATA 0 TO ADDR 4
   ```

7. Enter SHOW PROBE to see the captured responses. The display should read:

   ```
   SHOW PROBE CAPTURED RESPONSES
   SIG= 1    ASYNC LEVEL = 1 0
   COUNT = 1    CLOCKED LEVEL = 1
   ```

   The stimulus, WRITE DATA 0 TO ADDR 4, drove A2 high. PodSync, caused by a pod valid address cycle, clocked this event which then becomes CLOCKED LEVEL history. The ASYNC LEVEL captured a high and a low between the ARM and the SHOW. To illustrate this further perform the following steps.

8. Arm the probe:

   ```
   ARM PROBE FOR CAPTURE USING SYNC
   ```

9. Place probe tip on logic common.

10. Place probe tip on +5 V.

11. Place probe tip on address line A1.

12. Enter the following commands:

```
WRITE DATA 0 TO ADDR 2
```

13. Enter SHOW PROBE to see the captured responses. The display should read:

```
SHOW PROBE CAPTURED RESPONSES
SIG =     ASYNC LEVEL = 1X0
COUNT =     CLOCKED LEVEL = 1
```

The value of COUNT varies depending on how often you probe a point or remove a probe from that point.

The CLOCKED LEVEL displayed the level at each clock pulse (in this case one PodSync pulse) after the last SHOW command. The probe only saw one such clocked level (high). The ASYNC LEVEL showed all states the probe tip had seen after the last time SHOW was executed.

Probe Level (Externally Synchronized)

This example shows how to use the externally clocked level history functions of the probe using ARM PROBE and SHOW PROBE. This example illustrates the use of the clock module in conjunction with the probe.

1. Enter the command:

```
RUN UUT STARTING ADDR F8800 BREAK F8000
```

2. Fill the video RAM with data by entering the command:

```
RUN UUT STARTING ADDR F8A00 BREAK F8000
```

3. Make the following connections on the video controller using the clock module (connections on the Demo/Trainer UUT are shown in parentheses):

```
START (green)   - VSYNC  (U72-18)
STOP (red)      - VSYNC  (U72-18)
CLOCK (yellow)  - CCLK   (U72-16)
ENABLE (blue)   - BLANK  (U72-17)
COMMON (black)  - Any convenient UUT common
```

4. Set the synchronization as follows:

```
SYNC PROBE TO EXT ENABLE LOW CLOCK ↓ ...
... START ↓ STOP ↓
```

5. Arm the probe for measurement:

```
ARM PROBE FOR CAPTURE USING SYNC
```

6. On the Demo/Trainer UUT, place the probe on the U75 pins listed in Step 7.

7. Display the captured responses:

```
SHOW PROBE CAPTURED RESPONSES

pin 2      ASYNC LEVEL = 1 0
           CLOCKED LEVEL = 1 0

pin 7      ASYNC LEVEL = 1 0
           CLOCKED LEVEL = 1 0

pin 10     ASYNC LEVEL = 1 0
           CLOCKED LEVEL = 1 0

pin 15     ASYNC LEVEL = 1 0
           CLOCKED LEVEL = 1 0
```

Using the Probe Lights

This example shows the differences between the operation of the lights for clocked mode and the non-clocked mode. In the clocked mode, the lights are latched to the last clocked input. In the non-clocked mode, the lights follow the static input state and have built-in stretchers for observing rapid transitions.

The probe lights can be used to detect an open data line that cannot be identified by the BUS test.

1. Open data bit 8. On the Demo/Trainer UUT, open SW1-5.

2. Start BUS test at address 0. No fault will be detected.

3. Enter the command:

   ```
   TEST RAM FAST ADDR 0 TO 1FFFE DATA ...
   ... MASK FFFF ADDR STEP 2 DELAY 250 SEED 0
   ```

4. The display should read:

   ```
   TEST RAM FAST ADDR 0 TO 1FFFE DATA ...
   .. MASK FFFF ADDR STEP 2 DELAY 250 SEED 0
   attempted to write data 0 at 0 read 100
   data line D8 stuck high
   ```

5. Synchronize the probe:

   ```
   SYNC PROBE TO POD DATA
   ```

6. Arm the probe for capture:

   ```
   ARM PROBE FOR CAPTURE USING SYNC
   ```

7. Probe the output pin of the data buffer (U23-11).

8. Perform the following stimulus:

```
TOGGLE DATA 0 MASKED BY 100 ADDR 0
```

Probe indicator results: red ON and green ON.

9. Arm the probe for capture.

```
ARM PROBE FOR CAPTURE USING SYNC
```

10. Probe D8 (U41-14) on the RAM chip.

11. Perform a stimulus:

```
TOGGLE DATA 0 MASKED BY 100 ADDR 0
```

Probe indicator results: red OFF, yellow ON, and green OFF.

The open circuit is diagnosed as being between the RAM chip and the output of the data buffer.

I/O Module (Pin Level)

An I/O module can measure levels on several pins of a UUT component. In this example, data is written to port A, which is one of three input/output ports (A, B, and C) on the Demo/Trainer UUT. The levels on each of port A's eight pins can then be measured by the I/O module.

Ports A, B, and C are provided by U31 an 8255 PIA. U31 is a 40-pin DIP IC. A 40-pin clip module should be used, and its pin numbers map to the same pin numbers on the I/O module: Pin 4 on the I/O module maps to pin 4 on the clip module, and so on.

To write data to port A using the pod and then read it at Port A using the I/O module:

1. Attach the 40-pin clip module to the I/O module. Clip U31 on the Demo/Trainer UUT.

2. Initialize the 8255 PIA to "mode 0", all ports output:

```
WRITE DATA 80 TO ADDR 4006
ADDR OPTION:  I/O BYTE
```

3. Write data AA to port A:

```
WRITE DATA AA TO ADDR 4000
ADDR OPTION:  I/O BYTE
```

4. To read each port bit, enter the command:

```
INPUT I/O MOD 1 PIN <x> LEVEL
```

Where "x" is the I/O module pin listed below:

I/O Module Pin	*Port A Pin*	*Level*
4	PA0	0
3	PA1	1
2	PA2	0
1	PA3	1
40	PA4	0
39	PA5	1
38	PA6	0
37	PA7	1

I/O Module (Word Level)

In the previous example, we had to enter the command "INPUT I/O MOD 1 PIN <x> LEVEL", eight times in order to see the levels at each of Port A's eight pins. If we define those pins as a WORD, we could enter one command to see the levels on all eight pins at once.

A WORD is labeled set of I/O module pins, ordered so as to map in a known way to clip module pins, and hence to the pins of the UUT component being clipped over.

This example is very similar to the previous one: data will be written to the eight pins of port A on an 8255 PIA (U31 on the Demo/Trainer UUT). However, we will then define those pins as a word and use INPUT I/O MOD WORD to read the levels at all the pins as a hex number.

The sequence is as follows:

1. Attach the 40-pin clip module to the I/O module. Clip U31 on the Demo/Trainer UUT. U31 is an 8255 PIA whose Port A pins are now the same as WORD 1.

2. Initialize the 8255 to "mode 0", all ports output:

   ```
   WRITE DATA 80 TO ADDR 4006
   ADDR OPTION:  I/O BYTE
   ```

3. Write data AA (10101010) to port A:

   ```
   WRITE DATA AA TO ADDR 4000
   ADDR OPTION:  I/O BYTE
   ```

4. Now we must define our pin order. We will look at port A and define a word that maps the port A bits as follows:

   ```
              MSB                           LSB

   Port A :   PA7 PA6 PA5 PA4 PA3 PA2 PA1 PA0
   I/O Mod:    37  38  39  40   1   2   3   4
   ```

5. To define the word, enter the command:

   ```
   SET I/O MOD 1 WORD 5 AS PINS 37 38 39 ...
   ... 40 1 2 3 4 0
   ```

6. Read the data word at port A by entering the command:

   ```
   INPUT I/O MOD 1 WORD 5
   ```

7. The input data is displayed as a hex number, the same one written in step 3:

   ```
   INPUT I/O MOD 1 WORD 5 = AA
   ```

I/O Module Data Comparison 6.3.6.

The SET I/O MOD WORD COMPARE command can be used to verify that a particular bit pattern has occurred at chosen input pins of an I/O module. The user can therefore detect states on components that cannot be directly read by the microprocessor, such as output-only latches.

The compare condition shows as a message to the user and is available as an output on the side of the I/O module. When the state is detected (as a hex number), the event is reported on the display. In addition, the DCE (data compare equal) output at the side of the I/O module becomes active.

The example below illustrates the data comparison capability. We start by defining (as a word) the I/O module pins connected to Port A on U31, the PIA. We then program the comparison register with a chosen data bit pattern which will be looked for on the pins of the defined word. Next we initialize the PIA as in the previous examples. Lastly, we write the chosen data pattern to Port A, and a system interrupt results from recognition of the pattern by the comparison circuit.

1. Define a word by entering the command:

```
SET I/O MOD 1 WORD 5 AS PINS 37 38 39 ...
... 40 1 2 3 4 0
```

2. Initialize the 8255 to "mode 0", all ports output:

```
WRITE DATA 80 TO ADDR 4006
ADDR OPTION: I/O BYTE
```

3. Attach the 40-pin clip module to the I/O module. Clip U31 on the Demo/Trainer UUT.

4. Up to 40 pins on one I/O module can be specified for comparison as a 10-digit hex number. To sense the pattern 10101010 on the pins, enter the command:

```
SET I/O MOD 1 COMPARE WORD 5 TO AA
```

NOTE

The comparison occurs at all I/O module pins. If the defined word is not 40 bits long, the comparison only occurs at pins defined in SET I/O MOD WORD. Extra (more significant) bits implied by SET I/O MOD COMPARE (step 2) are ignored.

5. Write data AA (10101010) to Port A:

```
WRITE DATA AA TO ADDR 4000
ADDR OPTION: I/O BYTE
```

6. The pre-set pattern is sensed, an interrupt occurs, and the system will report:

```
attempted to write at 4000
compare condition reached in I/O module 1
```

7. Once a data-compare-equal (DCE) condition is reached, the feature must be reset by entering the SET I/O MOD COMPARE command again if you desire to continue comparing the defined word to the same bit pattern.

Driving Nodes to Known Levels 6.3.7.

Both the probe and the I/O module can be used to drive nodes during troubleshooting or as stimuli to test circuits that cannot be stimulated by the pod. Both can overdrive nodes that are stuck, as long as the nodes are not tied to Vcc or Common. The probe can synchronize its drive to pod access cycles (address or data), external signals, the freerun clock, or static levels. The I/O module can drive latched or pulsed levels on all pins simultaneously.

Freerun Probe Output

Freerun pulsing is used as a stimulus to inputs that need to be asynchronous with respect to the microprocessor or other UUT clock signals. On the Demo/Trainer UUT it could be used to test inputs to the interrupt priority encoder as follows:

1. Using the SETUP MENU key, enter:

   ```
   SETUP POD REPORT INTR ACTIVE ON
   ```

2. Set probe to freerun pulsing mode.

   ```
   OUTPUT PROBE PULSER LOW
   ```

3. Enter the command:

   ```
   READ ADDR 0
   ADDR OPTION:  MEMORY WORD
   ```

4. Press the LOOP key. The display should be something like the following:

   ```
   READ ADDR 0 = FFFF
   ```

   The data read can vary depending on the UUT.

5. On the Demo/Trainer UUT, probe U20-4. The system should display the following if the interrupt priority encoder chip is working properly:

   ```
   READ ADDR 0 = FFFF
   attempted to READ at 0
   interrupt INTR pod pin 57 is active
   ```

Probe Output (Pod Synchronized)

Synchronized output is primarily used to stimulate nodes during the microprocessor read and write cycles. This is useful when troubleshooting open or stuck bus lines beyond the UUT bus buffers.

For example, assume a RAM failure on data bit D0. We need to determine whether the path from the RAM back to the microprocessor is good. To do this:

1. Enter the command:

   ```
   SYNC PROBE TO POD DATA
   ```

2. Enter the command:

   ```
   OUTPUT PROBE PULSER TOGGLE
   ```

3. Enter the command:

   ```
   READ ADDR 0
   ```

4. Press the LOOP key. The display should be:

   ```
   READ ADDR 0 = <x>
   ```

   Where "x" is the hex data read on the RAM data lines.

5. Use the probe to trace D0 back from the RAM chip data bit (U55-14) through the bus buffer (U3-11 and U3-9) to the microprocessor (U14-36). If the path is good, the value of "x" should change indicating that the line is being toggled by the probe.

Probe Output (Externally Synchronized)

1. Enter the following command:

```
RUN UUT STARTING ADDR F8800 BREAK F8000
```

2. Fill the video RAM with data by entering the command:

```
RUN UUT STARTING ADDR F8A00 BREAK F8000
```

3. Make the following connections on the video controller using the clock module (connections on the Demo/Trainer UUT are shown in parentheses):

```
START (green)   - VSYNC   (U72-18)
STOP (red)      - VSYNC   (U72-18)
CLOCK (yellow)  - CCLK    (U72-16)
ENABLE (blue)   - BLANK   (U72-17)
COMMON (black)  - Any convenient UUT common
```

4. Set the synchronization as follows:

```
SYNC PROBE TO EXT ENABLE LOW CLOCK ↓ ...
... START ↓ STOP ↓
```

5. Connect a CRT monochrome monitor to the video output connector (J3) on the Demo/Trainer UUT.

6. Enter the following command:

```
OUTPUT PROBE PULSER TOGGLE
```

7. Arm the probe for capture:

```
ARM PROBE FOR CAPTURE USING SYNC
```

8. Press the LOOP key. The BUSY light should begin flashing on and off.

9. Probe the pin at U78-2 while simultaneously viewing the display on the monitor (vertical bars).

I/O Module Pin Output

1. Attach a 20-pin clip to the "A" side of the I/O module. Clip U3, the lower-byte data bus buffer of the Demo/Trainer UUT.

   We would like to drive data bit D7 high. D7 is on pin 18 of the data buffer. I/O module pins map one-to-one to pins of a 20-pin clip module connected to the "A" side of an I/O module. To drive pin 18 on the IC, we drive pin 18 of the I/O module.

2. Drive data bit D7 high:

   ```
   OUTPUT I/O MOD 1 PIN 18 DATA HIGH LATCH
   ```

3. The results can be verified using the pod to read the data (assuming all other bits are set to 0):

   ```
   READ ADDR 0 = 80
   ```

4. To remove the level from pin 18, set the pin to the high-impedance state.

   ```
   OUTPUT I/O MOD 1 PIN 18 3-STATE
   ```

   To drive any of the other lower order data bus pins, merely change the PIN number in the OUTPUT command. In practice it is easier to use the WORD mode if you work with more than one pin at a time.

I/O Module Word Output

In this example, one I/O module, one 20-pin clip module, and one 16-pin clip module are used. We need to test a 16-bit data bus that uses two buffers, one each for the lower and upper bytes. The problem is that the buffers are 20-pin components. A closer look at the components reveals that we can test them simultaneously using a 20-pin clip on one component and a 16-pin clip on the other.

The 20-pin clip module is connected to side "A" and is mapped to the I/O module and the IC as shown below:

IC Function:	D7	D6	D5	D4	D3	D2	D1	D0
IC Pin:	18	17	16	15	14	13	12	11
Clip Mod Pin:	18	17	16	15	14	13	12	11
I/O Mod Pin:	18	17	16	15	14	13	12	11

The 16-pin clip module is connected to side "B" and is mapped to the I/O module and the IC as shown below:

IC Function:	D15	D14	D13	D12	D11	D10	D9	D8
IC Pin:	18	17	16	15	14	13	12	11
Clip Mod Pin:	16	15	14	13	12	11	10	9
I/O Mod Pin:	40	39	38	37	36	35	34	33

1. Define I/O module WORD 1 so the 16-pin clip (side "B") is the upper data byte and the 20-pin clip (side "A") is the lower byte. To do this, enter the command:

   ```
   SET I/O MOD 1 WORD 1 AS PINS 40 39 38 ...
   ... 37 36 35 34 33 18 17 16 15 14 13 ...
   ... 12 11 0
   ```

2. Connect the 20-pin clip (side "A") to U3 and the 16-pin clip (side "B") to U23. Pin 8 on the smaller clip should be aligned with pin 10 of U23.

3. Output the data word A5A5 to the I/O module:

```
OUTPUT I/O MOD 1 WORD 1 DATA A5A5 LATCH
```

4. Verify that the correct data was written by performing a pod read.

```
READ ADDR 0 DATA = A5A5
ADDR OPTION:  MEMORY WORD
```

5. Turn the drive off and return the WORD 1 pins to the high-impedance state by entering the command:

```
OUTPUT I/O MOD 1 WORD 1 3-STATE MASK FFFF
```

Section 7
General Maintenance

This section contains the following general maintenance information for the 9100A/9105A:

- Periodic maintenance and cleaning.
- Changing of line-voltage settings and fuses.
- Shipping.
- Service support.

PERIODIC MAINTENANCE AND CLEANING 7.1.

You must periodically clean the fan filter element and the micro-floppy disk drives. Other cleaning can be done as desired for the sake of appearance and convenience of use.

Cleaning the Fan Filter 7.1.1.

The fan filter should be cleaned at least every 90 days, or more often as necessary, to ensure free passage of cooling air. The filter is behind the louvered filter cover on the right side of the 9100A/9105A mainframe. To remove the filter, pull outwards on the cover at either side of the upper disk drive until the latching pins snap out of the chassis. Then lift the cover up until the bottom is free. Remove the foam filter from the cover and clean it with warm water and detergent.

Cleaning the Floppy Disk Drive(s) 7.1.2.

The floppy disk drive(s) should be cleaned at least one a year. Clean a drive by running a commercially available cleaning disk in the drive for 5 seconds.

General Cleaning 7.1.3.

CAUTION

Do not use aromatic hydrocarbons or chlorinated solvents for cleaning. These may damage the plastic materials used in the instrument.

Avoid using excessive amounts of liquid, particularly around the keypad, keyboard, or disk drives.

The operator's display and the monitor screen should be cleaned with a soft cloth that has been slightly dampened with a cleaner. Use a commercially available lens or CRT cleaner or a non-abrasive household cleaner.

Clean the exterior and accessory cables with a mild solution of detergent and water or with a non-abrasive household cleaner.

The operator's keypad and programmer's keyboard should be gently cleaned with a cloth or towel slightly dampened with a non-abrasive household cleaner or a mild solution of detergent and water.

LINE VOLTAGE SETTINGS AND FUSES 7.2.

CAUTION

Before changing any line-voltage settings or fuses, the 9100A/9105A mainframe and the monitor ***must*** *be switched OFF.*

This section shows how to change the line voltage settings on the 9100A/9105A mainframe and the monitor. Also covered are how to change the 9100A/9105A mainframe line fuse, the probe fuse, the clock module fuse, and the I/O module fuse.

Changing the Mainframe Line Voltage

The VOLTAGE switch on the 9100A/9105A rear panel must be set to match the local line voltage; if the voltage setting is changed, the power fuse should be changed to the correct rating for the new voltage, as shown in the following table.

Voltage Switch Setting	*Line Voltage Range*	*Power Fuse (F1)*
110 V	90-130 VAC	2 A slow blow
220 V	180-264 VAC	1 A slow blow

Turn the voltage switch on the rear panel to the desired setting (110V or 220V). Check the power fuse rating and replace the fuse if necessary.

Changing the Monitor Line Voltage 7.2.2.

The voltage switch on the Fluke monochrome monitor's rear panel must be set to match the local line voltage; the monitor does not require a power fuse change when changing the line voltage setting.

Turn the voltage switch on the rear panel to the desired setting (110V or 220V). The Fluke monochrome monitor does not have a user-replaceable power fuse.

Changing the Mainframe Power Fuse 7.2.3.

The power fuse holder is on the rear panel and is labeled "F1". To remove the fuse, press in on the fuse holder cap and turn it counter-clockwise. Pull the fuse cap and fuse straight out, and remove the fuse from the fuse cap.

See the table on the previous page for the correct fuse rating for each line voltage.

The monochrome monitor does not have a user-replaceable fuse.

Changing the Probe Fuse 7.2.4.

The probe fuse protects the probe and mainframe against high-current shorts to common caused by incorrect connection of the probe common lead to the UUT. If the probe fuse blown message appears on the operator's display, disconnect the probe common lead from the UUT and determine the why the fuse blew before replacing it.

The PROBE FUSE holder is on the right side panel of the mainframe, next to the PROBE connector. To remove the fuse, press in on the fuse holder cap and turn it counter-clockwise. Pull the fuse cap and fuse straight out, and remove the fuse from the fuse cap.

Use a 0.25A, 250V fast blow fuse.

Changing the Clock Module Fuse 7.2.5.

The clock module fuse protects the clock module and mainframe against high-current shorts to common caused by incorrect connection of the clock module COMMON lead to the UUT. If the clock module fuse blown message appears on the operator's display, disconnect the clock module COMMON lead from the UUT and determine why the fuse blew before replacing it.

The clock module fuse holder is on the clock module. To remove the fuse, use the same procedure as for the probe fuse.

Use a 0.25A, 250V fast blow fuse.

Changing the I/O Module Fuse 7.2.6.

The I/O module fuse protects the I/O module and mainframe against high-current shorts to common caused by incorrect connection of the I/O module COMMON lead to the UUT. If the I/O module fuse blown message appears on the operator's display, disconnect the COMMON lead of the indicated I/O module from the UUT and determine why the fuse blew before replacing it.

The I/O module fuse holder is on the back of the I/O module, near the cable. To remove the fuse, use the same procedure as for the probe fuse.

Use a 1.0A, 250V slow blow fuse.

SHIPPING 7.3.

If it becomes necessary to ship the 9100A/9105A, use its original foam-packed shipping container to prevent unnecessary damage. If the original packaging is not available, order new containers from your Fluke sales representative.

When shipping or moving a 9100A, be sure the hard disk drive is parked to prevent damage to the internal hard disk or mechanical drive components. The hard disk drive will park itself after a period of 10 seconds without a hard-disk access. The DISK ACCESS status light will flash and a beep will sound as the disk drive parks.

Before shipping a 9100A or a 9105A, each floppy disk drive should have a special cardboard insert placed into it. If these inserts have not been saved, a floppy disk may be inserted instead.

SERVICE SUPPORT 7.4.

Each Fluke 9100A/9105A is warranted as shown on the inside front cover of this manual.

Factory authorized service for the 9100A/9105A system is available worldwide at various locations. See Appendix G for a complete list of these service centers.

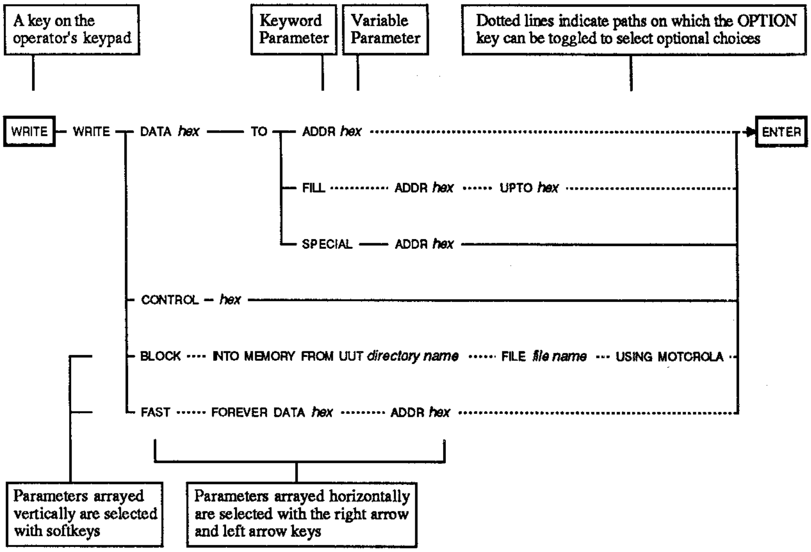
A service kit that troubleshoots a failed 9100A/9105A down to the component level is available. This kit includes:

- The *9100A Service Manual.*
- A service utility program disk.
- An extender board.
- TL/1 GFI troubleshooting software.

The service kit requires a 68000 pod and a second 9100A (or 9105A) with an I/O module to do the troubleshooting. The 9100A service kit is available from Fluke through the Service Parts Department. Order Fluke stock number 818948.

The *9100A Service Manual* is available by itself. It contains information such as required equipment, theory of operation, maintenance, repair, lists of replaceable parts, schematic diagrams, and option and accessories information. The *9100A Service Manual* does not include troubleshooting procedures. Order Fluke stock number 809210.

Appendix A
Keypad-Syntax Quick Guide


A key on the operator's keypad
Keyword Parameter
Variable Parameter
Dotted lines indicate paths on which the OPTION key can be toggled to select optional choices
WRITE
WRITE
DATA hex
TO
ADDR hex
ENTER
FILL
ADDR hex
UPTO hex
SPECIAL
ADDR hex
CONTROL
hex
BLOCK
INTO MEMORY FROM UUT directory name
FILE file name
USING MOTOROLA
FAST
FOREVER DATA hex
ADDR hex
Parameters arrayed vertically are selected with softkeys
Parameters arrayed horizontally are selected with the right arrow and left arrow keys

Read diagrams from left to right. Horizontal parameters are selected with the right arrow and left arrow keys. Vertical parameters are selected with softkeys.

Variables Parameters

························ A decimal number that you type on the keypad. The softkeys INC (increment) and DEC (decrement) appear when this field is selected.

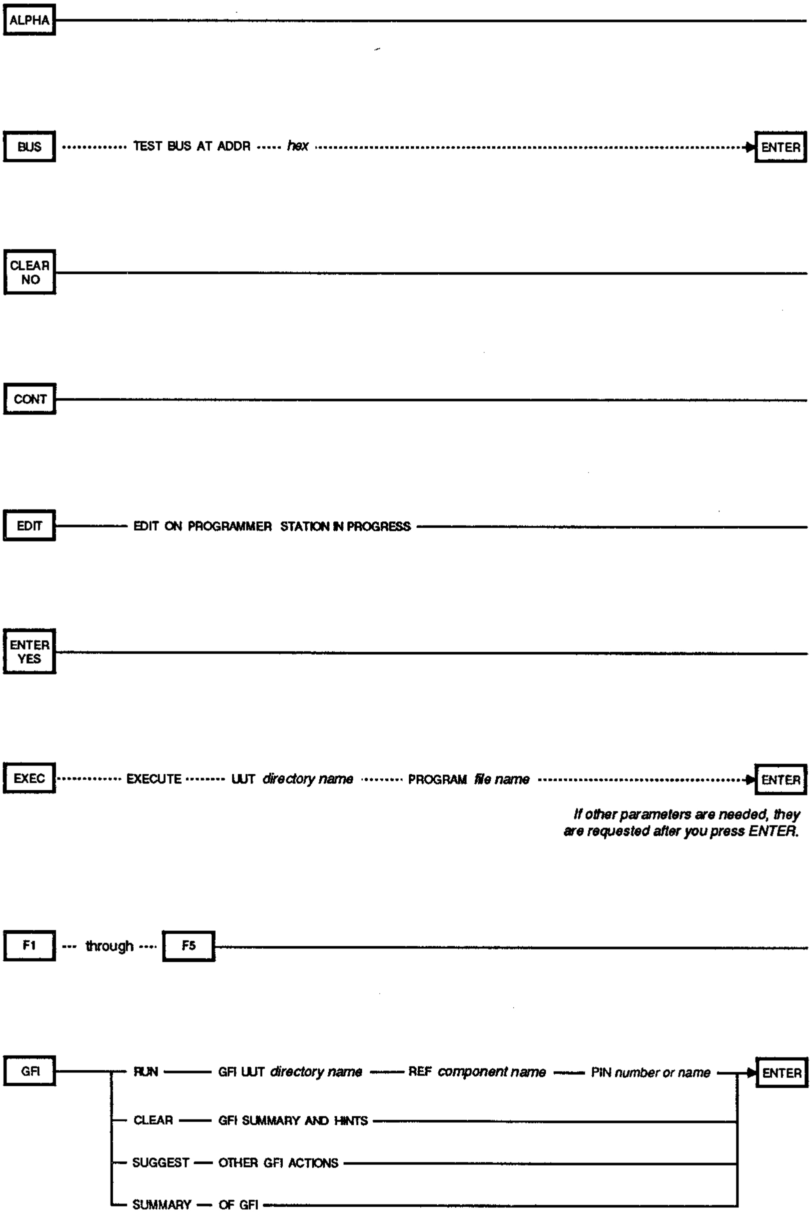
hex ························ A hexadecimal number that you type on the keypad. The softkeys INC (increment) and DEC (decrement) appear when this field is selected.

file name ················ An alphanumeric name of a program or text file (e.g., "Test5").

component name ···· An alphanumeric name of a component (e.g., "U55").

directory name ········ An alphanumeric name of a file directory for a specific type of UUT (e.g., "DEMO").

pod name ················ An alphanumeric name of a pod (e.g., "80286").



ALPHA

BUS ············ TEST BUS AT ADDR ····· *hex* ········→ ENTER

CLEAR NO

CONT

EDIT —— EDIT ON PROGRAMMER STATION IN PROGRESS

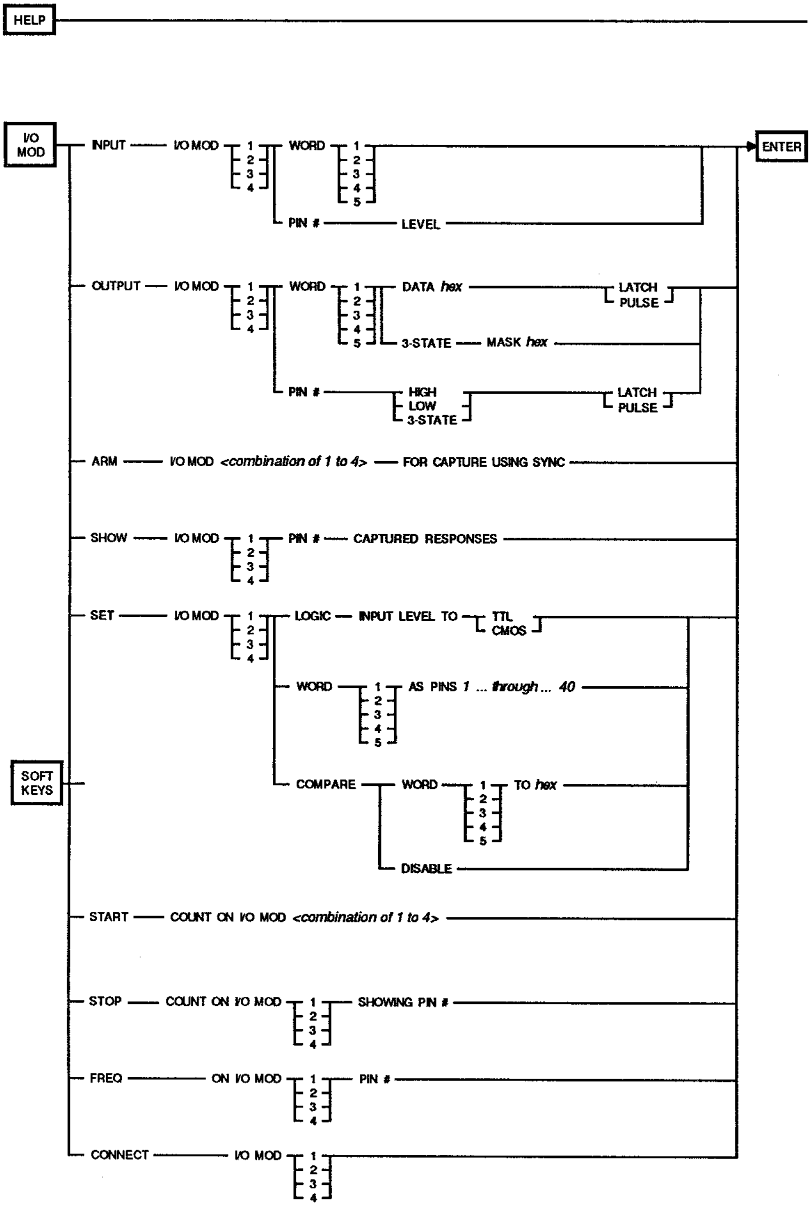
ENTER YES

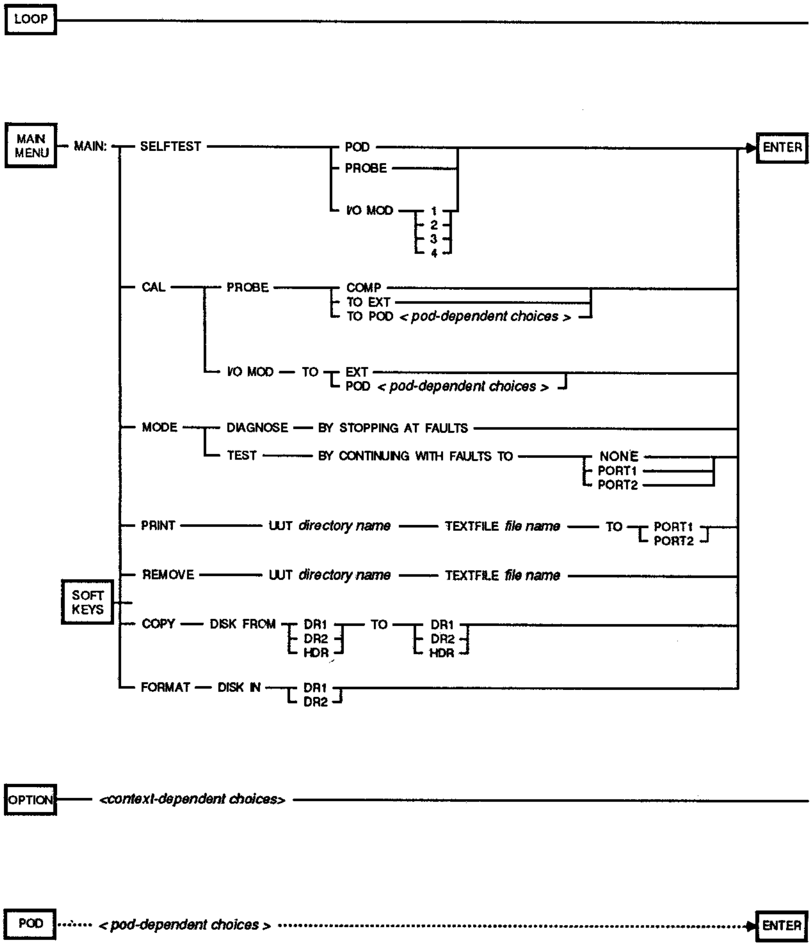
EXEC ············ EXECUTE ········ UUT *directory name* ········ PROGRAM *file name* ········→ ENTER

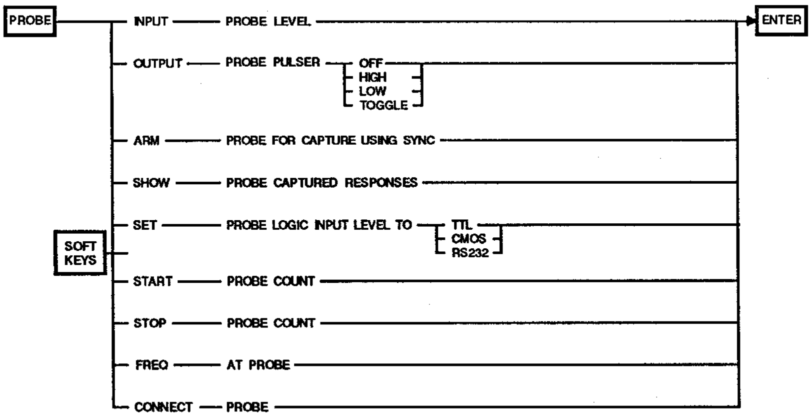
If other parameters are needed, they are requested after you press ENTER.

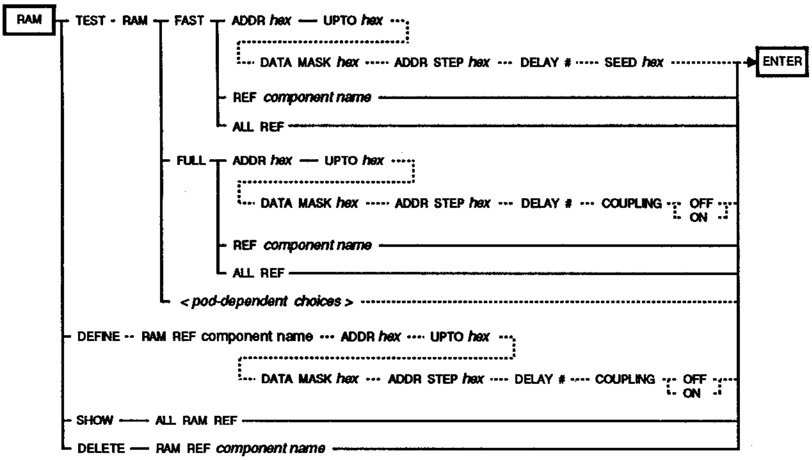
F1 --- through ---- F5

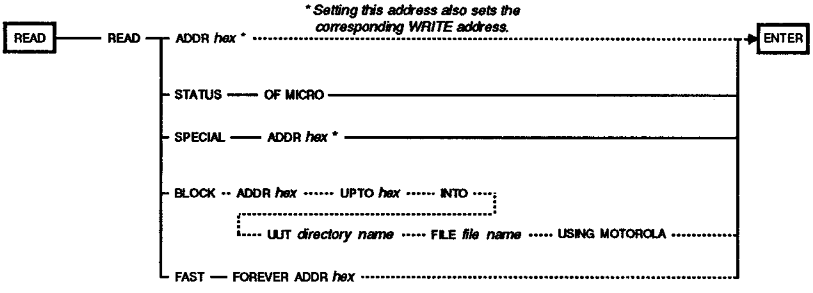
GFI
- RUN —— GFI UUT *directory name* —— REF *component name* —— PIN *number or name* → ENTER
- CLEAR —— GFI SUMMARY AND HINTS
- SUGGEST —— OTHER GFI ACTIONS
- SUMMARY —— OF GFI

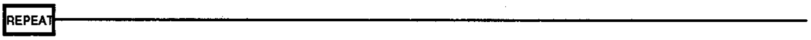

HELP
I/O MOD
INPUT — I/O MOD 1 2 3 4 — WORD 1 2 3 4 5
PIN # — LEVEL
ENTER
OUTPUT — I/O MOD 1 2 3 4 — WORD 1 2 3 4 5 — DATA hex — LATCH PULSE
3-STATE — MASK hex
PIN # — HIGH LOW 3-STATE — LATCH PULSE
ARM — I/O MOD <combination of 1 to 4> — FOR CAPTURE USING SYNC
SHOW — I/O MOD 1 2 3 4 — PIN # — CAPTURED RESPONSES
SET — I/O MOD 1 2 3 4 — LOGIC — INPUT LEVEL TO — TTL CMOS
WORD — 1 2 3 4 5 — AS PINS 1 ... through... 40
COMPARE — WORD — 1 2 3 4 5 — TO hex
DISABLE
SOFT KEYS
START — COUNT ON I/O MOD <combination of 1 to 4>
STOP — COUNT ON I/O MOD 1 2 3 4 — SHOWING PIN #
FREQ — ON I/O MOD 1 2 3 4 — PIN #
CONNECT — I/O MOD 1 2 3 4

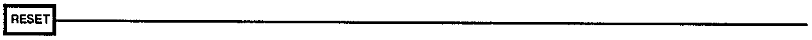

LOOP
MAIN MENU
MAIN:
SELFTEST
POD
PROBE
I/O MOD
1
2
3
4
ENTER
CAL
PROBE
COMP
TO EXT
TO POD < pod-dependent choices >
I/O MOD
TO
EXT
POD < pod-dependent choices >
MODE
DIAGNOSE
BY STOPPING AT FAULTS
TEST
BY CONTINUING WITH FAULTS TO
NONE
PORT1
PORT2
PRINT
UUT directory name
TEXTFILE file name
TO
PORT1
PORT2
REMOVE
UUT directory name
TEXTFILE file name
SOFT KEYS
COPY
DISK FROM
DR1
DR2
HDR
TO
DR1
DR2
HDR
FORMAT
DISK IN
DR1
DR2
OPTION
<context-dependent choices>
POD
< pod-dependent choices >
ENTER

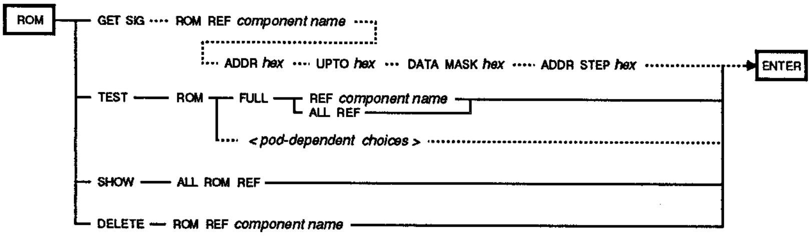

PROBE
SOFT KEYS
INPUT — PROBE LEVEL
OUTPUT — PROBE PULSER — OFF / HIGH / LOW / TOGGLE
ARM — PROBE FOR CAPTURE USING SYNC
SHOW — PROBE CAPTURED RESPONSES
SET — PROBE LOGIC INPUT LEVEL TO — TTL / CMOS / RS232
START — PROBE COUNT
STOP — PROBE COUNT
FREQ — AT PROBE
CONNECT — PROBE
ENTER

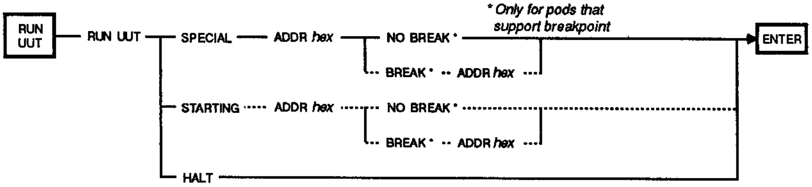

RAM
TEST - RAM
FAST — ADDR *hex* — UPTO *hex*
DATA MASK *hex* ····· ADDR STEP *hex* ··· DELAY # ····· SEED *hex*
REF *component name*
ALL REF
FULL — ADDR *hex* — UPTO *hex*
DATA MASK *hex* ····· ADDR STEP *hex* ··· DELAY # ··· COUPLING — OFF / ON
REF *component name*
ALL REF
< *pod-dependent choices* >
DEFINE ·· RAM REF component name ··· ADDR *hex* ···· UPTO *hex*
DATA MASK *hex* ··· ADDR STEP *hex* ···· DELAY # ···· COUPLING — OFF / ON
SHOW — ALL RAM REF
DELETE — RAM REF *component name*
ENTER


* Setting this address also sets the corresponding WRITE address.
READ
READ — ADDR *hex* *
STATUS — OF MICRO
SPECIAL — ADDR *hex* *
BLOCK ·· ADDR *hex* ······ UPTO *hex* ······ INTO
UUT *directory name* ····· FILE *file name* ····· USING MOTOROLA
FAST — FOREVER ADDR *hex*
ENTER

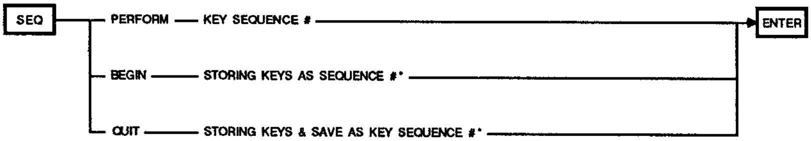




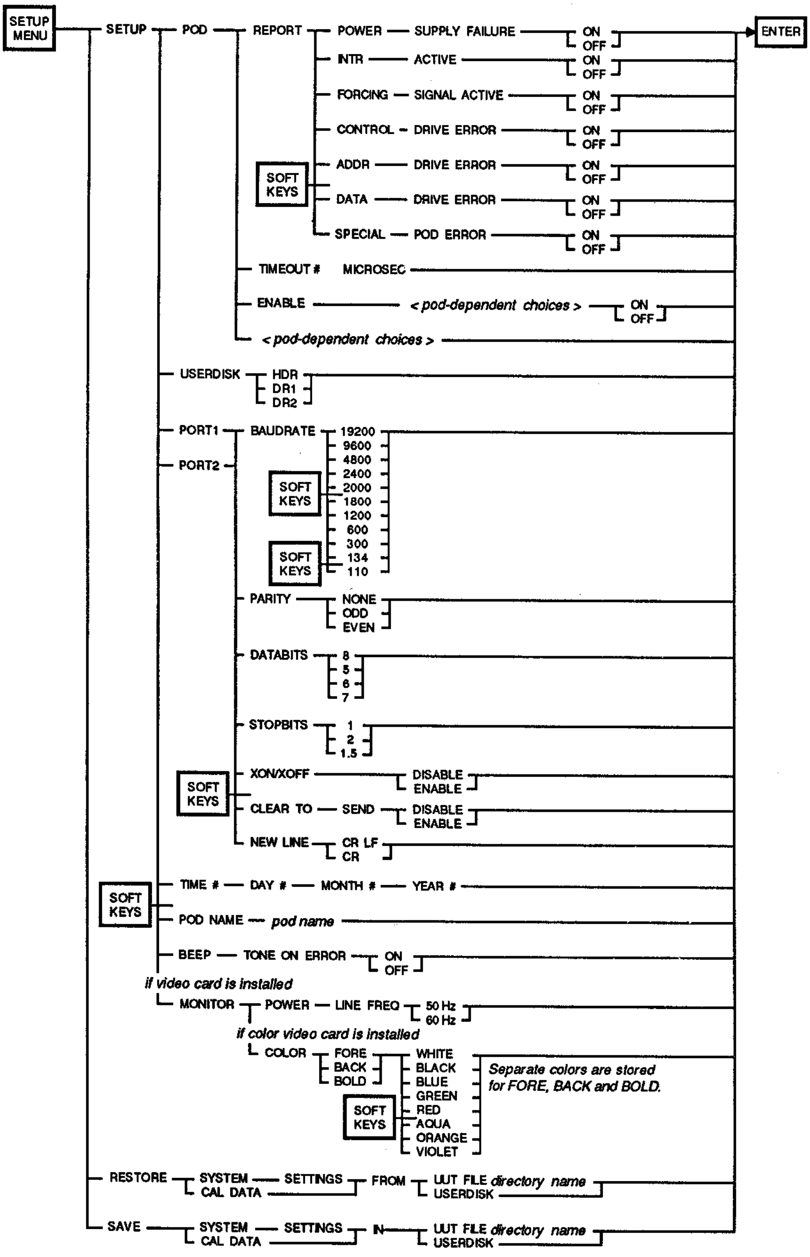


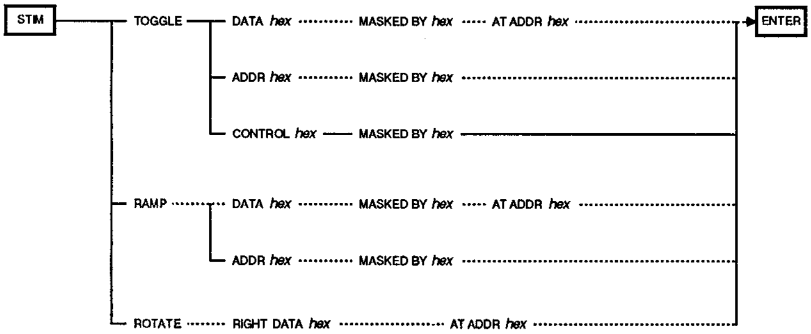


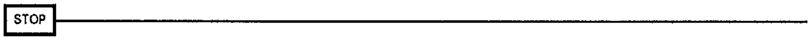
** Only for pods that support breakpoint*

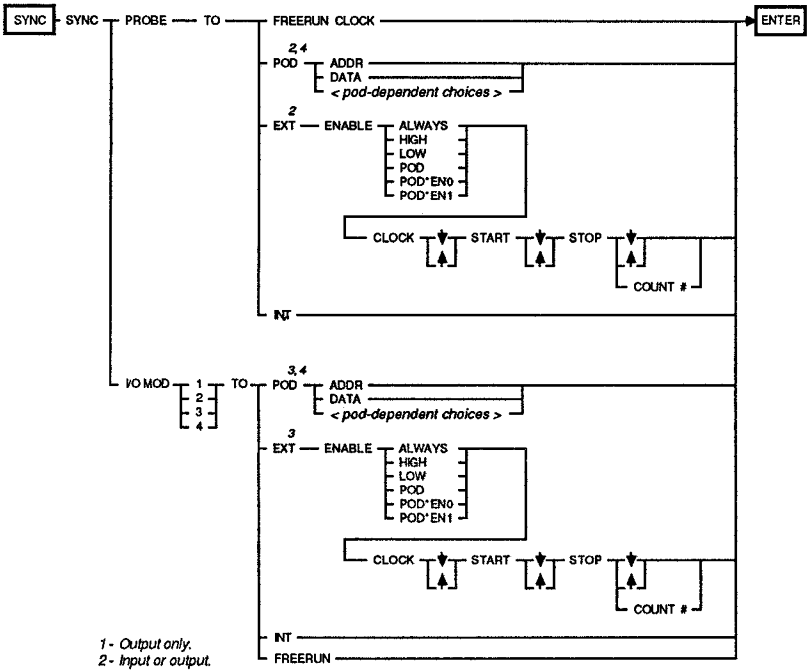

SEQ
PERFORM
KEY SEQUENCE #
ENTER
BEGIN
STORING KEYS AS SEQUENCE #*
QUIT
STORING KEYS & SAVE AS KEY SEQUENCE #*

** Whatever you enter after pressing ENTER is recorded as the key sequence, until you press the SEQ key, the QUIT softkey, and the ENTER key, in that order.*

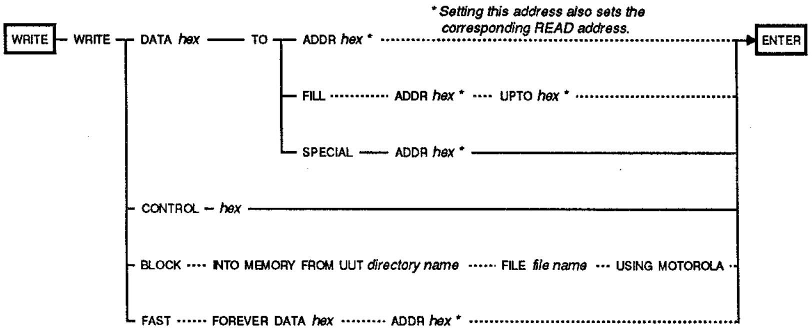

SETUP MENU
SETUP
POD
REPORT
POWER — SUPPLY FAILURE — ON / OFF
INTR — ACTIVE — ON / OFF
FORCING — SIGNAL ACTIVE — ON / OFF
CONTROL — DRIVE ERROR — ON / OFF
ADDR — DRIVE ERROR — ON / OFF
DATA — DRIVE ERROR — ON / OFF
SPECIAL — POD ERROR — ON / OFF
SOFT KEYS
ENTER
TIMEOUT # MICROSEC
ENABLE — <pod-dependent choices> — ON / OFF
<pod-dependent choices>
USERDISK — HDR / DR1 / DR2
PORT1
PORT2
BAUDRATE — 19200 / 9600 / 4800 / 2400 / 2000 / 1800 / 1200 / 600 / 300 / 134 / 110
SOFT KEYS
SOFT KEYS
PARITY — NONE / ODD / EVEN
DATABITS — 8 / 5 / 6 / 7
STOPBITS — 1 / 2 / 1.5
XON/XOFF — DISABLE / ENABLE
CLEAR TO — SEND — DISABLE / ENABLE
NEW LINE — CR LF / CR
SOFT KEYS
TIME # — DAY # — MONTH # — YEAR #
POD NAME — pod name
BEEP — TONE ON ERROR — ON / OFF
SOFT KEYS
if video card is installed
MONITOR — POWER — LINE FREQ — 50 Hz / 60 Hz
if color video card is installed
COLOR — FORE / BACK / BOLD — WHITE / BLACK / BLUE / GREEN / RED / AQUA / ORANGE / VIOLET
SOFT KEYS
Separate colors are stored for FORE, BACK and BOLD.
RESTORE — SYSTEM — SETTINGS / CAL DATA — FROM — UUT FILE directory name / USERDISK
SAVE — SYSTEM — SETTINGS / CAL DATA — IN — UUT FILE directory name / USERDISK







1 - Output only.
2 - Input or output.
3 - Input only.
4 - Setting the probe synchronization sets the corresponding I/O module synchronization, and vice versa.


* Setting this address also sets the corresponding READ address.
WRITE
WRITE
DATA *hex*
TO
ADDR *hex* *
FILL
ADDR *hex* *
UPTO *hex* *
SPECIAL
ADDR *hex* *
CONTROL – *hex*
BLOCK
INTO MEMORY FROM UUT *directory name*
FILE *file name*
USING MOTOROLA
FAST
FOREVER DATA *hex*
ADDR *hex* *
ENTER

Appendix B
I/O Module Clip/Pin Mapping

Clip size = 14, module installed on "A" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	1	14	=	20
2	=	2	13	=	19
3	=	3	12	=	18
4	=	4	11	=	17
5	=	5	10	=	16
6	=	6	9	=	15
7	=	7	8	=	14

Clip size = 14, module installed on "B" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	21	14	=	40
2	=	22	13	=	39
3	=	23	12	=	38
4	=	24	11	=	37
5	=	25	10	=	36
6	=	26	9	=	35
7	=	27	8	=	34

Clip size = 16, module installed on "A" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	1	16	=	20
2	=	2	15	=	19
3	=	3	14	=	18
4	=	4	13	=	17
5	=	5	12	=	16
6	=	6	11	=	15
7	=	7	10	=	14
8	=	8	9	=	13

Clip size = 16, module installed on "B" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	21	16	=	40
2	=	22	15	=	39
3	=	23	14	=	38
4	=	24	13	=	37
5	=	25	12	=	36
6	=	26	11	=	35
7	=	27	10	=	34
8	=	28	9	=	33

Clip size = 18, module installed on "A" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	1	18	=	20
2	=	2	17	=	19
3	=	3	16	=	18
4	=	4	15	=	17
5	=	5	14	=	16
6	=	6	13	=	15
7	=	7	12	=	14
8	=	8	11	=	13
9	=	9	10	=	12

Clip size = 18, module installed on "B" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	21	18	=	40
2	=	22	17	=	39
3	=	23	16	=	38
4	=	24	15	=	37
5	=	25	14	=	36
6	=	26	13	=	35
7	=	27	12	=	34
8	=	28	11	=	33
9	=	29	10	=	32

Clip size = 20, module installed on "A" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	1	20	=	20
2	=	2	19	=	19
3	=	3	18	=	18
4	=	4	17	=	17
5	=	5	16	=	16
6	=	6	15	=	15
7	=	7	14	=	14
8	=	8	13	=	13
9	=	9	12	=	12
10	=	10	11	=	11

Clip size = 20, module installed on "B side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	21	20	=	40
2	=	22	19	=	39
3	=	23	18	=	38
4	=	24	17	=	37
5	=	25	16	=	36
6	=	26	15	=	35
7	=	27	14	=	34
8	=	28	13	=	33
9	=	29	12	=	32
10	=	30	11	=	31

Clip size = 24, module installed on "A" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	1	24	=	20
2	=	2	23	=	19
3	=	3	22	=	18
4	=	4	21	=	17
5	=	5	20	=	16
6	=	6	19	=	15
7	=	7	18	=	14
8	=	8	17	=	13
9	=	9	16	=	12
10	=	10	15	=	11
11	=	29	14	=	32
12	=	30	13	=	31

Clip size = 24, module installed on "B" side

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	21	24	=	40
2	=	22	23	=	39
3	=	23	22	=	38
4	=	24	21	=	37
5	=	25	20	=	36
6	=	26	19	=	35
7	=	27	18	=	34
8	=	28	17	=	33
9	=	29	16	=	32
10	=	30	15	=	31
11	=	9	14	=	12
12	=	10	13	=	11

Clip size = 28

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	1	28	=	40
2	=	2	27	=	39
3	=	3	26	=	38
4	=	4	25	=	37
5	=	5	24	=	36
6	=	6	23	=	35
7	=	7	22	=	34
8	=	8	21	=	33
9	=	9	20	=	32
10	=	10	19	=	31
11	=	11	18	=	30
12	=	12	17	=	29
13	=	13	16	=	28
14	=	14	15	=	27

Clip size = 40

Clip		*I/O Mod Pin*	*Clip*		*I/O Mod Pin*
1	=	1	40	=	40
2	=	2	39	=	39
3	=	3	38	=	38
4	=	4	37	=	37
5	=	5	36	=	36
6	=	6	35	=	35
7	=	7	34	=	34
8	=	8	33	=	33
9	=	9	32	=	32
10	=	10	31	=	31
11	=	11	30	=	30
12	=	12	29	=	29
13	=	13	28	=	28
14	=	14	27	=	27
15	=	15	26	=	26
16	=	16	25	=	25
17	=	17	24	=	24
18	=	18	23	=	23
19	=	19	22	=	22
20	=	20	21	=	21

Appendix C
Pod-Related Information

INTRODUCTION C.1.

The *Supplemental Pod Information For 9100A/9105A Users Manual* provides the following information for each pod:

- **Address space options:** Shows the parameter names, parameter values, and all legal combinations of parameter values. Address space options are accessed through the OPTIONS key at the operator's keypad and display or through the TL/1 getspace command.
- **Pod-specific set-up information:** Shows the parameters that are available through the SETUP MENU key on the operator's keypad and display or through the TL/1 podsetup command. These parameters are used to set-up the pod for a specific UUT.
- **TL/1 support programs:** Shows a list of the TL/1 programs that are available in the pod library. These programs provide convenient interfacing with any special functions built into the pod.
- **Pod sync calibration and offset data:** Shows the UUT signal, active (reference) edge of the signal, and the default offset from the specified reference edge for each sync mode.

POD CALIBRATION AND OFFSETS C.2.

Calibration is the process by which the internal delay lines in the I/O module and probe are adjusted to correctly align (in time) the clock and signals to be sampled. To calibrate an I/O module or probe to a pod for a particular pod sync mode, you are prompted to probe a signal on the UUT. The specified reference edge on that signal is found by adjusting the delay lines in the I/O module or probe relative to the internal PodSync signal. The appropriate delay, labeled "tcal" in Figure C-1, may vary from one pod to another and from one sync mode to another. If calibration is not performed, then a default setting is used for the tcal value. When calibration is performed, the measured value for tcal replaces the default value.

Once the reference edge is found, then an offset is applied to that edge to determine just where in time the I/O module or probe will latch data.

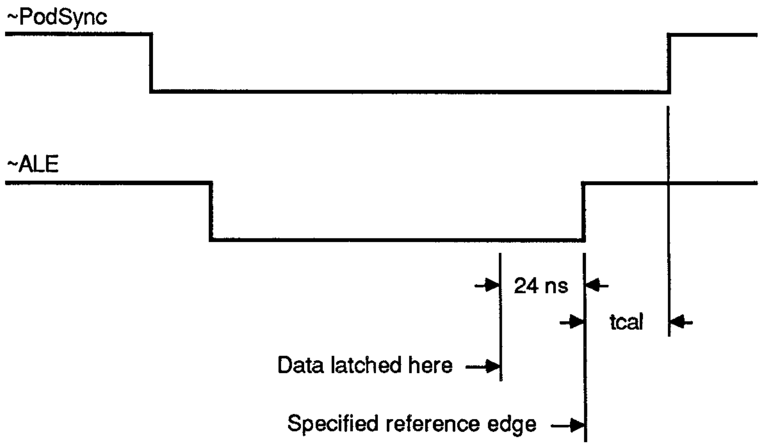
The following is an example for an imaginary "xyz" pod showing how the offset data is listed in this appendix and how this data would apply to real waveforms. Pod calibration and offset data for the "xyz" pod would appear in this appendix as follows:

Sync Mode	*UUT Signal*	*Edge of Signal*	*Offset from Edge*
ADDR	~ALE	rising	-24 ns

In the imaginary "xyz" pod, the reference edge for address sync is the rising edge of the ~ALE (address latch enable) signal. Furthermore, the offset data shows that a valid address is best captured when sampled 24 nanoseconds before the rising edge of ~ALE. (A positive offset would have indicated that the address should be latched after the reference edge.)

The waveforms corresponding to the above example are shown below The "~" symbol indicates that both the ALE signal and the PodSync signal are active low.

As a result of the calibration process, the offset value is set to the default value for the sync mode in use.



Appendix D
Standard Equipment, Options, and Accessories

9100A STANDARD EQUIPMENT

9100A/SYS 115V
9100A/SYS 230V

Digital Programming Test System

- 9100A (115V or 230V) Digital Test System
- 9100A-004 (115V or 230V) Programmer's Station
- 9100A-003 I/O Module
- Y9100A-DCS DIP Clip Module Set

9100A 115V
9100A 230V

Digital Test System

- 9100A (115V or 230V) mainframe
- Probe with accessories
- Clock module with accessories
- Four manuals
  Getting Started
  Automated Operations Manual
  Technical User's Manual
  Applications Manual
- System Disk 1
- System Disk 2
- Master User Disk
- Y8091 Ten 3.5-inch micro-floppy diskettes

9105A STANDARD EQUIPMENT

9105A 115V Digital Test Station
9105A 230V

- 9105A (115V or 230V) mainframe
- Probe with accessories
- Clock module with accessories
- Four manuals
  - *Getting Started*
  - *Automated Operations Manual*
  - *Technical User's Manual*
  - *Applications Manual*
- System Disk 1
- System Disk 2
- Master User Disk 1
- Master User Disk 2
- Y8091 Ten 3.5-inch micro-floppy diskettes

9100A (only) OPTIONS

9100A-004 115V* Programmer's Station
9100A-004 230V*

- Monochrome (115V or 230V) monitor
- Two manuals
  - *Programmer's Manual*
  - *TL/1 Reference Manual*
- Video card (monochrome configuration)
- Video cable
- Programmer's keyboard
- Programmer's System Disk
- 2M-byte additional RAM

* If you purchase this option separately (for example, after you have purchased a 9100A/9105A system), you must return your system to a Fluke Service Center for installation of the newly purchased option.

9100A-005 115V* 9100A-005 230V*	Programmer's Station (less monochrome monitor) - Two manuals *Programmer's Manual* *TL/1 Reference Manual* - Video card (color configuration) - Programmer's keyboard - Programmer's System Disk - 2M-byte additional RAM

9105A (only) OPTIONS

9105A-008*	Real-time clock

9100A/9105A OPTIONS

9000A-010	Demo/Trainer UUT
9100A-003	I/O Module - I/O module - Y9100A-20L 20-lead Flying Lead Module - Calibration module
9100A-009 115V* 9100A-009 230V*	Video, Monochrome - Monochrome (115V or 230V) monitor - Video card (monochrome configuration) - Video cable
9100A-011*	Video card (color configuration)
9100A-016	Hard Disk Backup Utility
9100A-019	2M Memory Expansion

* If you purchase this option separately (for example, after you have purchased a 9100A/9105A system), you must return your system to a Fluke Service Center for installation of the newly purchased option.

ACCESSORIES

Y8091	Ten 3.5-inch micro-floppy diskettes
Y9100A-14D	14-pin DIP Clip Module
Y9100A-16D	16-pin DIP Clip Module
Y9100A-18D	18-pin DIP Clip Module
Y9100A-20D	20-pin DIP Clip Module
Y9100A-24D	24-pin DIP Clip Module
Y9100A-28D	28-pin DIP Clip Module
Y9100A-40D	40-pin DIP Clip Module
Y9100A-14S	14-pin SO Clip Module
Y9100A-16S	16-pin SO Clip Module
Y9100A-20S	20-pin SO Clip Module
Y9100A-24S	24-pin SO Clip Module
Y9100A-28S	28-pin SO Clip Module
Y9100A-20P	20-pin PLCC Clip Module
Y9100A-28P	28-pin PLCC Clip Module
Y9100A-20L	20-lead Flying Lead Module (with 25 clips)
Y9100A-DCS	DIP Clip Module Set - Y9100A-14D 14-pin DIP Clip Module - Y9100A-16D 16-pin DIP Clip Module - Y9100A-18D 18-pin DIP Clip Module - Y9100A-20D 20-pin DIP Clip Module - Y9100A-24D 24-pin DIP Clip Module - Y9100A-28D 28-pin DIP Clip Module - Y9100A-40D 40-pin DIP Clip Module

Y9100A-100	Card Edge Fixture Kit - Card Edge Fixture Kit - Card Edge Fixture Manual
Y9100A-101	Card Edge Fixture Base
Y9100A-102	Card Edge Interface Module
Y9100A-7201	Probe Accessories Kit - 1 ea. Probe Tip - 1 ea. Probe Tip Cable Assembly - 1 ea. Ground Clip Hook (Large) - 1 ea. Ground Cable Assembly - 8 ea. Clock Module Clips (Small)
Y9100A-7203	Foot Switch

Appendix E
9100A/9105A Specifications

ELECTRICAL SPECIFICATIONS

Probe

Input Threshold:

Logic Level	*TTL Voltage*	*CMOS Voltage*	*RS-232 Voltage*
1	2.6 to 5.0 V	3.7 to 5.0 V	3.2 to 30 V
1 or X	2.2 to 2.6 V	3.3 to 3.7 V	2.8 to 3.2 V
X	1.0 to 2.2 V	1.2 to 3.3 V	-2.8 to 2.8 V
X or 0	0.6 to 1.0 V	0.8 to 1.2 V	-3.2 to -2.8 V
0	0.0 to 0.6 V	0.0 to 0.8 V	-30 to -3.2 V

Input Impedance:

70 KΩ shunted by less than 33 pF

Data Timing for Synchronous Measurements:

Max frequency	40 MHz
Min pulse width (high or low)	12.5 nsec
Min pulse width (3-state)	20.0 nsec
Setup Times:	
- Data to Clock	5 nsec
- Start, Stop, or Enable to Clock	10 nsec
Hold Time:	
- Clock to Enable	10 nsec
- Clock to Start or Stop	0 nsec

Data Timing for Asynchronous Measurements:

Max frequency	40 MHz
Min pulse width (high or low)	12.5 nsec
Min pulse width - Invalid (X):	
- TTL or CMOS	100 nsec ± 20 nsec
- RS-232	2000 nsec ± 400 nsec

Transition Counting:

Max frequency	40 MHz
Max count	16777215 (+ overflow)
Max stop count	65535 clocks

Frequency Measurement:

Max frequency	40 MHz
Resolution	20 Hz
Accuracy	± 250 ppm ± 20 Hz

Output Pulser:

High	> 3.5 V @ 200 mA for less than 10 µsec @ 1% duty cycle > 4.0 V @ 5 mA continuously
Low	< 0.8 V @ 200 mA for less than 10 µsec @ 1% duty cycle < 0.4 V @ 5 mA continuously

Clock Module

Input Thresholds (all lines):	1.6 V ± 0.2 V
Input Impedance:	50 KΩ shunted by less than 10 pF

Clock, Start, Stop, and Enable Input Speed:

Max repetition rate	40 MHz
Min pulse width	12.5 nsec

RS-232 Interfaces

One connector isolated (system-referenced), the other connector non-isolated (earth-referenced).

Baud rates	110, 134, 300, 600, 1200, 1800, 2400,4800, 9600, 19200
Parity	Odd, even, or none
Data bits	5, 6, 7, or 8
Stop bits	1, 1.5, or 2
XON/XOFF (Ctrl-S/Ctrl-Q)	Disable/Enable
Clear-to-Send	Disable/Enable
New line	Carriage Return and Line Feed (CRLF) or Carriage Return (CR)

I/O Module

Data Output :

Current (> 10 msec)	± 200 mA ± 10%
Current (< 10 msec)	± 2 A ± 10%
Pattern rate (One module driven)	approximately 35 kHz
Pattern depth (One module driven during 10 msec high current pattern drive mode)	256 patterns
Max current (at $V_{out} \geq 2$ V) (per pin, driving high)	250 mA

Max current (at $V_{out} \leq 0.8$ V) (per pin, driving low)	150 mA

Input Thresholds:

Logic Level	*TTL Voltage*	*CMOS Voltage*
1	2.6 to 5.0 V	3.4 to 5.0 V
1 or X	2.1 to 2.6 V	2.9 to 3.4 V
X	1.0 to 2.1 V	1.2 to 2.9 V
X or 0	0.6 to 1.0 V	0.8 to 1.2 V
0	0.0 to 0.6 V	0.0 to 0.8 V

Input Impedance

50 KΩ minimum, shunted by less than 80 pF

Clock, Start, Stop, and Enable Inputs:

Logic Thresholds	
- Low	0.8 V maximum
- High	2.0 V minimum
Input Current	± 1 μA

Input/Output Overvoltage Protection:

± 15 V for one minute maximum, any pin, one at a time

Transition Counting:

Max frequency	10 MHz
Max count (Transition mode)	8388607 counts (+overflow)
Frequency accuracy (Frequency mode)	± 250 ppm ± 2 Hz

Stop Counter:

Max frequency	10 MHz
Max count	65535 clocks

Clock:

Max frequency	10 MHz
Min pulse width	50 nsec

Data Timing for Synchronous Measurements:

Max frequency of clock	10 MHz
Data setup time	30 nsec
Data hold time	30 nsec
Min pulse width (Start/Stop/Enable/Clock)	50 nsec
Start edge setup time (Before clock edge, for clock edge to be recognized)	0 nsec
Stop edge hold time (After clock edge, for clock edge to be recognized)	10 nsec
Enable setup time (Before clock edge, for clock edge to be recognized)	0 nsec
Enable hold time (After clock edge, for clock edge to be recognized)	10 nsec

Data Timing for Asynchronous Measurements:

Max frequency	10 MHz
Min pulse width (high or low)	50 nsec
Min pulse width (3-state)	150 nsec

Data Compare Equal (DCE):

Min pulse width (Data and Enable)	75 nsec

General

Line Voltage:

90 to 132 VAC, 47 to 440 Hz
180 to 264 VAC, 47 to 63 Hz

Power Consumption:

- Mainframe	150 W maximum
- Monitor	50 W maximum

Safety:

Designed to meet ANSI/UL 478, IEC 348, IEC 435, and CSA 556B standards.

PHYSICAL SPECIFICATIONS

Operating Temperature:

5 to 27° C, 95% RH maximum (non-condensing).

27 to 40° C, RH decreasing linearly from 95% to 50% (non-condensing).

Programmer's Station:

24-line by 80-column CRT monitor with video card installed in mainframe. 87-key keyboard with separate cursor-control, and hardkey and softkey function keys.

Storage/Shipping Temperature:

-20 to 60° C, 8% to 80% RH, (non-condensing). Micro-floppy media limited from 5 to 60° C, 8% to 80% RH (non-condensing).

Size:

Mainframe:	14.0 x 34.3 x 50.8 cm (H x W x D) (5.5 x 13.5 x 20.0 in)
Monitor:	30.5 x 33.5 x 33.0 cm (H x W x D) (12.0 x 13.2 x 13.0 in)
ASCII Keyboard:	5.0 x 47.2 x 21.2 cm (H x W x D) (2.0 x 18.5 x 8.3 in)

Weight:

Mainframe:	8.3 kg (18.2 lb)
Monitor:	11.1 kg (24.5 lb)
ASCII Keyboard:	1.6 kg (3.5 lb)

Appendix F
Fault Messages

Fault messages are displayed when the 9100A/9105A detects a fault on the UUT or when certain conditions are detected in the 9100A/9105A system. The fault messages are grouped as follows:

- Bus Test Fault Messages
- ROM Test Fault Messages
- RAM Test Fault Messages
- General Pod-Related Fault Messages
- Memory Interface Pod Fault Messages
- Other Fault Messages

When a fault message appears on the operator's display, you can press the HELP key on the operator's keypad to access additional information about UUT faults that might cause this fault message. The HELP information appears on the operator's display, and you can use the Up Arrow and Down Arrow keys to scroll through the information.

BUS TEST FAULT MESSAGES F.1.

The following fault messages can result from a BUS test:

```
addr line <name> pod pin <pin> stuck high
```

The address line named <name> located at pod pin number <pin> is stuck high.

```
addr line <name> pod pin <pin> stuck low
```

The address line named <name> located at pod pin number <pin> is stuck low.

```
addr line <name> pod pin <pin> tied
```

The address line named <name> located at pod pin number <pin> is tied to some other line.

```
data line <name> pod pin stuck high
```

The data line named <name> located at pod pin number <pin> is stuck high.

```
data line <name> pod pin <pin> stuck low
```

The data line named <name> located at pod pin number <pin> is stuck low.

```
data line <name> pod pin <pin> tied
```

The data line named <name> located at pod pin number <pin> is tied to some other line.

ROM TEST FAULT MESSAGES F.2.

The following fault messages can result from a ROM test:

```
addr line <name1> tied to addr line <name2>
```

The address line named <name1> is tied to the address line named <name2>.

```
addr line <name> stuck
```

The address line named <name> is stuck either high or low.

```
all ROM data bits stuck high
```

The data read from ROM has a signature corresponding to all data bits being a "1".

```
all ROM data bits stuck low
```

The data read from ROM has a signature corresponding to all data bits being a "0".

```
data line <name1> tied to data line <name2>
```

The data line named <name1> is tied to the data line named <name2>.

```
data line <name> stuck low
```

or

```
data line <name> struck high
```

The data line named <name> is stuck low (or high, if that form of the fault message appears).

```
read incorrect sig <data1> expected <data2>
```

The signature read, <data1>, is different from the specified correct signature, <data2>.

RAM TEST FAULT MESSAGES F.3.

The following fault messages can result from a RAM test:

```
addr line <name1> pod <pin1> tied to addr line
     <name2> pod <pin2>
```

The address line named <name1> located at pod pin number <pin1> is tied to the address line named <name2> located at pod pin number <pin2>.

```
addr line <name1> tied to data line <name2>
```

The address line named <name1> may be tied to the data line named <name2>.

```
addr line <name1> may be tied to data line <name2>
```

The address line named <name1> may be tied to the data line named <name2>, but other causes of failure are also likely.

```
addr line <name> stuck or open
```

The address line named <name> is stuck or open.

```
memory cell for <name1> coupled to memory cell for
     <name2>
```

A memory cell associated with the line named <name1> is coupled to a memory cell associated with the line named <name2>.

```
memory cell for <name> stuck high
```

A memory cell associated with the line named <name> is stuck high.

```
memory cell for <name> stuck low
```

A memory cell associated with the line named <name> is stuck low.

```
data line <name1> pod <pin1> tied to data line
     <name2> pod <pin2>
```

The data line named <name1> located at pod pin number <pin1> is tied to the data line named <name2> located at pod pin number <pin2>.

```
cannot modify RAM data
```

After writing two different data patterns to the same memory location, the same incorrect data was read after both writes.

```
data line <name> pod pin <pin> stuck high
```

The data line named <name> located at pod pin number <pin> is stuck high.

```
data line <name> pod pin <pin> stuck low
```

The data line named <name> located at pod pin number <pin> is stuck low.

```
read incorrect data <data1> expected <data2>
```

The 9100A/9105A encountered an error in the data read from RAM, but could not further localize the fault.

```
RAM data retention fault (bad refresh?)
```

After all attempts to isolate a fault at a failed memory location, data is written to the failed address and reread to verify that the data is correctly stored. Following a 10 second pause, the data read back from the failed address is different than the data that is originally written.

GENERAL POD-RELATED FAULT MESSAGES F.4.

The following fault messages can result from faults or conditions detected by the pod:

```
addr line <name> pod pin <pin> not drivable
```

The actual logic level on the address line named <name> located at pod pin number <pin> did not match the logic level that the pod tried to assert.

```
bad UUT power supply
```

The UUT power supply had an incorrect power supply voltage level.

```
breakpoint reached
```

The pod detected that the program executing in the UUT reached the condition that the pod was programmed to detect. The pod then terminated the execution of the UUT program.

```
control line <name> pod pin <pin> not drivable
```

The actual logic level on the control line named <name> located at pod pin number <pin> did not match the logic level that the pod tried to assert.

```
data line <name> pod pin <pin> not drivable
```

The actual logic level on the data line named <name> located at pod pin number <pin> did not match the logic level that the pod tried to assert.

```
enabled line <name> pod pin <pin> causes timeout
```

The pod did not respond to the 9100A/9105A because the line named <name> located at pod pin number <pin> did not relinquish control of the microprocessor bus.

```
forcing signal <name> pod pin <pin> is active
```

The pod detected activity on the forcing signal named <name> located at pod pin number <pin>. The SET MENU key on the operator's keypad can be used to stop the reporting of this activity, if required.

```
interrupt <name> pod pin <pin> is active
```

The pod detected activity on the interrupt signal named <name> located at pod pin number <pin>. The SET MENU key on the operator's keypad can be used to stop the reporting of this activity, if required.

```
pod timeout bad UUT clock
```

The pod did not respond to the 9100A/9105A because it did not detect a UUT clock signal.

```
pod timeout bad UUT power supply
```

The pod did not respond to the 9100A/9105A due to an incorrect UUT power supply voltage level.

```
pod timeout recovered
```

The pod did not respond to the 9100A/9105A when requested, but a subsequent reset to the pod re-established communications.

```
pod selftest code = <number>
```

The pod failed the specified selftest. Refer to the Fluke pod manual for the pod you are using.

```
read incorrect data <data1> expected <data2>
```

The pod encountered an error in the data read from the UUT. It read <data1> when it expected to read <data2>.

```
setup causes pod timeout
```

The pod did not respond to the 9100A/9105A because of an incorrect setting in the SETUP MENU.

```
<name> fault pod <pin>
```

The pod detected a fault on the line named <name> located at pod pin number <pin>. You may need to refer to the Fluke pod manual for the microprocessor you are using for more information about this line.

```
<pod special message>
```

The pod-special fault messages are specific to each type of pod. Refer to the Fluke pod manual for the pod you are using.

MEMORY INTERFACE POD FAULT MESSAGES F.5.

The following fault messages can result from faults or conditions dectected by the interface pod:

```
pod buscycle CLK BAD
      check status line <name> <pin>
      check control line <name> <pin>
      check line <name> <pin>
```

The internal clock used by the 9132A pods is called the bus cycle clock. This clock is generated by the sync module, from a combination of UUT signals. The signals referred to by the error messages generate the bus cycle clock. Either the connections from these lines to the sync module, or the lines themselves are bad.

```
kernel fault
      <message1>
      <message2>
```

A failure is detected by the 9132 TEST BUS, but was not diagnosed.

```
ROM1 CS or OE is stuck invalid
      <message1>
      <message2>
```

The 9132 TEST BUS program caused the microprocessor to be reset. The program expected the microprocessor to fetch an instruction from the boot ROM by ROM module 1. This situation could result from an address decoding fault, or an open trace between the decoder and the boot ROM.

```
UUT clock <name> <pin> slow or stuck
```

The 9132 TEST BUS program sensed the clock signal being too slow. If the clock signal were stuck to one level, this fault would also be generated.

```
microprocessor stopped or bad
      <message1>
      <message2>
```

The 9132 TEST BUS program found that the microprocessor was not generating the expected control signals. The microprocessor could have executed a HALT instruction, the control lines could be stuck, or the microprocessor could be defective.

```
no µP reset detected on <name> <pin>
      <message1>
      <message2>
```

The 9132 TEST BUS program tried to reset the microprocessor, but no reset was detected at the microprocessor by the sync mode. Check connections between the sync mode and the UUT.

```
BAD reset address: <address> expected
      <name> <pin> was high
      <name> <pin> was low
```

The 9132 TEST BUS program caused the microprocessor to be reset, and the microprocessor generated the reset address. The address monitored by the boot ROM (by ROM module #1) is incorrect. The offending bits are reported as high or low.

```
BAD reset data: <data> expected
      <name> <pin> was high
      <name> <pin> was low
```

The 9132 TEST BUS program caused the microprocessor to be reset, and expected the microprocessor to fetch a byte of data from ROM Module 1. The byte of data, monitored by the Sync Module is incorrect. The lines that are correct are specified in the fault display.

```
Addr line <name> <pin> was high - expected low
Addr line <name> <pin> was low - expected high
```

The 9132 TEST BUS program the indicated address line(s) in a different state than was expected. The fault could be caused by a line stuck high or low, or tied to another line, (address or data).

OTHER FAULT MESSAGES. F.6.

The following additional fault messages can appear on the operator's display:

```
clock module fuse blown
```

The fuse for the clock module is open or missing.

```
compare condition reached in I/O module <number>
```

The specified I/O module (or modules) detected the combination of signal levels the 9100A/9195A was programmed to detect.

```
I/O module overcurrent fault
```

The current limit of an I/O module pin was exceeded.

```
I/O module <number> fuse blown
```

The fuse for the I/O module number <number> is open or missing.

```
probe fuse blown
```

The fuse for the probe is open or missing.

```
unknown or intermittent fault occurred
```

Some sort of fault occurred, but the 9100A/9105A was not able to determine exactly what it was. The fault is probably intermittent or transient.

Appendix G
Fluke Sales Offices and Technical Service Centers

U.S. Service Locations

California
Fluke Technical Center
16969 Von Karman Ave., Suite 100
Irvine, CA 92714
Tel: (714) 863-1723

Fluke Technical Center
46610 Landing Parkway
Fremont, CA 94538
Tel: (415) 651-5112

Colorado
Fluke Technical Center
14180 East Evans Ave.
Aurora, CO 80014
Tel: (303) 695-1000

Florida
Fluke Technical Center
940 N. Fern Creek Avenue
Orlando, FL 32803
Tel: (407) 896-4881

Illinois
Fluke Technical Center
1150 W. Euclid Ave.
Palatine, IL 60067
Tel: (312) 705-0500

Maryland
Fluke Technical Center
5640 Fishers Lane
Rockville, MD 20852
Tel: (301) 770-1576

New Jersey
Fluke Technical Center
E. 66 Midland Ave.
Paramus, NJ 07652
Tel: (201) 599-9500

Texas
Fluke Technical Center
1801 Royal Lane, Suite 307
Dallas, TX 75229
Tel: (214) 869-0311

Washington
Fluke Technical Center
1420 75th St SW
Everett, WA 98203
Tel: (206) 356-5560

International

Argentina
Coasin S.A.
Virrey del Pino 4071 DPTO E-65
1430 CAP FED
Buenos Aires, Argentina
Tel: 54 1 522-5248

Australia
Philips Customer Support
Scientific and Industrial
23 Lakeside Drive
Tally Ho Technology Park
East Burwood,
Victoria 3151 Australia

Philips Customer Support
Scientific & Industrial
25-27 Paul St. North/ PO Box 119
North Ryde N.S.W. 2113
Australia
Tel: 61 2 888-0423

Austria
Oesterreichische Philips
Industrie GmbH Technischer
Kundendienst
Unternehnmensbereich
Prof. System
Triesterstrasse 66
A-110 Austria
Tel: 43 222-60101

Belgium
Philips & MBLE Associated S.A.
Scientific & Industrial Div
80 Rue des deux Gares B-1070
Brussels, Belgium
Tel: 32 2 525 6111

Brazil
Hi-Tek Electronica Ltda.
Al. Amazonas 422 Alphaville,
CEP 06400 Barueri
Sao Paulo, Brazil
Tel: 55 11 421-5477

Canada
Fluke Electronics Canada Inc.
400 Brittania Rd. East, Unit #1
Mississauga, Ontario L4Z 1X9
Tel: 416-890-7600

Fluke Electronics Canada Inc.
7018 Cote de Liesse
St. Laurent, Quebec H4T 1E7
Canada
Tel: 731-8564

Fluke Electronics Canada Inc.
6815 8th St. N.E., Suite 135
Calgary, Alberta T2E 7H7
Canada
Tel: (403) 295-0822

Chile
Intronica Chile Ltda.
Casilla 16228
Santiago 9, Chile
Tel: 56 2 2321886

China, Peoples Republic of
Instrimpex - Fluke Service Center
(For Fluke products)
57, Xisi Dong Da Jie
Xicheng-qu
P.O. Box 9085
Beijing
People's Republic of China
Tel: 86 01 65-7281

Colombia
Sistemas E Instrumentacion, Ltda.
Carrera 13, No. 37-43, Of. 401
Ap. Aereo 29583
Bogota DE, Colombia
Tel: 57 232-4532

Denmark
Philips A/S
Technical Service I & E
Strandlodsvej 1A
PO Box 1919
DK-2300 - Copenhagen S.
Denmark
Tel: 45 1 572222

Ecuador
Proteco Coasin Cia., Ltda.
P.O. Box 228-AAve. 12 de Octubre
2285 y Orellana
Quito, Ecuador
Tel: 593 2 529684

Egypt and Sudan
Electronic Engineering
Liaison Office
P.O. Box 2891 Horreya
11361 Heliopolis
Cairo, Egypt
Tel: 202 695705

Finland
Instrumentarium Elektroniikka
P.O. Box 64, Vitikka 1
02631 ESPOO 63
Finland
Tel: 358-0-5281

France
S.A. Philips Industrielle
et Comerciale
Science et Industrie
105 Rue de Paris B.P. 62
93002 Bobigny Cedex
France
Tel: 33149428040

Germany (F.R.G.)
Philips GmbH
Unternehmensbereich Elektronik
fur Wissenschaft und Industrie
Service VSF
Oskar-Messter-Strasse 18
8045 Ismaning
West Germany
Tel: 49 89 9605-239

Great Britain
Philips Scientific Test
& Measurement Division
Colonial Way
Watford Hertfordshire WD2 4TT
United Kingdom
Tel: 44 923-240511

Greece
Philips S.A. Hellenique
11, Vrassida St.
Athens 612, Greece
Tel: 30 1 721140

Hong Kong
Schmidt & Co (H.K.) Ltd.
18th Fl., Great Eagle Centre
23 Harbour Road
Wanchai, Hong Kong
Tel: 852 58330-222

Technical Service Centers

India
Hinditron Services Pvt. Ltd
1st Floor, 17-B, Mahal
Industrial Estate
Mahakali Road Andheri East
Bombay 400 093, India
Tel: 91 22 6300043

Hinditron Services Pvt. Ltd.
33/44A Raj Maha Villas Extn.
8th Main Road
Bangalore 560 080, India
Tel: 91 812 363139

Hinditron Services Pvt. Ltd.
Field Service Center
Emerald Complex 1-7-264
5th Floor
114 Sarojini Devi Road
Secunderabad 500 003, India
Tel: 91 842 821117

Hindtron Services Pvt. Ltd.
15 Community Centre
Panchshila Park
New Delhi 110 017, India
Tel: 91 011 6433675

Indonesia
P.T. Lamda Triguna
P.O. Box 6/JATJG
Jakarta 13001, Indonesia
Tel: 62 21 8195365

Ireland
Philips Scientific
Test & Measurement Division
Colonial Way
Watford, Hertsfordshire
WD2 4TT, United Kingdom
Tel: 44 923 40511

Israel
R.D.T. Electronics Engineering Ltd.
P.O. Box 43137
Tel Aviv 61430
Israel
Tel: 972 3 483211

Italy
Philips S.p.A.
Sezione S & I, T & M Dept.
Viale Elvezia 2
20052 Monza, Italy
Tel: 39 39 3635 1

Japan
John Fluke Mfg. Co., Inc.
Japan Branch
Sumitomo Higashi Shinbashi Bldg.
1-1-11 Hamamatsucho
Minato-ku, Tokyo 105, Japan
Tel: 81 3 434-0181

Korea
Myoung Corporation
Yeo Eui Do P.O. Box 14
Seoul 150, Korea
Tel: 82 2 784-9942

Malaysia
Mecomb Malaysia Sdn. Bhd.
P.O. Box 24
46700 Petaling Jaya
Selangor, Malaysia
Tel: 60 3 774-3422

Mexico
Mexicana de Electronica
Industrial, S.A. (Mexel)
Diagonal No. 27
Entre Calle de Eugenia Y Ave
Colonia del Valle
C.P. 03100, Mexico
Tel: 90 5 660-4323

Netherlands
Nederlandse Philips
Badrijven B.V.
Building TQIII-4, P.O. Box 218
5600 MD Eindhoven
The Netherlands

Philips Nederland
Business Unit: Test & Meetapparaten
Hoerveneweg 55A
5017 AD Tilburg
Postbus 115, 5000 AC Tilburg
The Netherlands

New Zealand
Philips Customer Support
Scientific & Industrial Division
2 Wagener Place - PO Box 41021
Mt. Albert
Auckland, New Zealand
Tel: 65 4 735735

Norway
Morgenstierne & Co. A/S
Konghellegate 3
P.O. Box 6688, Rodelokka
Oslo 5, Norway
Tel: 47 2 356110

Pakistan
International Operations (PAK) Ltd.
505 Muhammadi House
I.I. Chundrigar Road
P.O. Box 5323
Karachi 2, Pakistan
Tel: 92 21 221127

Peru
Importaciones Y Representaciones
Electronicas S.A.
Avda. Franklin D. Roosevelt 105
Lima 1, Peru
Tel: 51 14 288650

Philippines
Spark Radio and Electronics Corp.
Greenhills, P.O. Box 610
San Juan, Metro-Manila 3113
Philippines
Tel: 63-2-775192

Portugal
Decada Espectral,
Equipamentos de Electronica e
Cientificos, SARL.
Av. Bombeiros Voluntario, Lote 102B
Miraflores/Alges
1495 Lisboa, Portugal
Tel: 351 1 410 3420

Singapore
Rank O'Connor's Singapore
(PTE) Ltd. O'Connor House
98 Pasir Panjang Road
Singapore 0511
Republic of Singapore
Tel: 65 4737944

South Africa
South African Philips
(Pty) Ltd. Service Dept.
195 Main Rd
Martindale, Johannesburg, 2092
South Africa
Tel: 27 11 470-5255

Spain
Equipos Y Sistemas S.A. (ESSA)
C/Apolanio Morales, 13B
Madrid 16, Spain
Tel: 4580150

Philips Iberica S.A.E.
Depto. Tecnico Instrumentacion I & E
C/Martinez Villergas 2
28027 Madrid, Spain
Tel: 34 1 402200

Sweden
Philips forsaljning AB
Customer Support
Tegeluddvagen 1
S-11584 Stockholm
Tel: 46 8 7821000

Switzerland
Philips AG Technischer Kundendienst
Allmendstrasse 140
Postfach 670
CH-8027 Zurich
Switzerland
Tel: 41 1 4882211

Taiwan
Schmidt Electronics Corp.
5th Floor, Cathay Min Sheng
Commercial Building
344 Min Sheng East Road
Taipei, Taiwan, R.O.C.
Tel: 886 2 5013468

Thailand
Measuretronix Ltd.
2102/63 Ramkamhaeng Rd.
Bangkok, 10240 Thailand
Tel: 66 2 3742516

Turkey
Erkman Elektronik Aletler
Ticaret Anonim Sirketi
Necatiby Cad 92/3
Karakoy, Istanbul, Turkey
Tel: 441546

Turk Philips Ticaret A.S.
Posta Kutusu 504 Beyoglu
Inonu Caddesi 78/80
Istanbul, Turkey
Tel: 90 1 1435910

Uruguay
Coasin Uruguaya S.A
Casilla de Correo 1400
Libertad 2529
Montevideo, Uruguay
Tel: 598 2 789015

Venezuela
Coasin C.A.
Calle 9 Con Calle 4
Edif. Edinurbi
Apartado de Correos
Nr-70 136 Los Ruices
Caracas 1070-A, Venezuela
Tel: 58 2 241-0309

International Sales Offices

Algeria
Bureau de Liaison Philips
24 rue Bougainville
El Mouradis, Alger
Tel: 60 14 05, TLX: 62221

Argentina
Coasin S.A.
Virrey del Pino 4071DPTO E-65
1430 CAP FED
Buenos Aires, Argentina
Tel: (54) (1) 552-5248
TLX: (390) 22284

Australia
Philips Scientific & Industrial Pty Ltd.
Centrecourt 25-27 Paul Street
North Ryde
Sydney N.S.W. 2113
Tel: (61) 88 88 222
FAX: (61) 2-888-0440

Austria
Osterreichische Philips Industrie GmbH
Geschaftsbereich I & E
Marktbereich Test-und MeBgerate
A-1101 WIEN, Triester StraBe 64
Tel: (0222) 60101/1722 DW

Bangladesh
Motherland Corporation
24 Hatkhola Road, Tikatuli
Dacca-3, Bangledesh
Tel: 257249, TLX: (950) 642022

Belgium
N.V. Philips Professional Systems S.A.
Test and Measurement
Tweestationsstraat 80
Rue des Deux Gares
Brussel 1070 Bruxelles
Tel: 2-5256111

Bolivia
Coasin Bolivia S.R.L.
Casilla 7295
La Paz, Bolivia
Tel: (591) (2) 40962
TLX: (336) 3233 COALAP BV

Brazil
ATP Hi-Tek Eletronica Ltda.
Al. Amazonas 422, Alphaville
Barueri 06400
Sao Paulo, Brazil
Tel: (55) (11) 421-5477
TLX: (391) 1171413

Brunei
Rank O'Connor's, Sdn Bhd
No. 8 Block D
Sufri Shophouse Complex
Mile 1 Jalan Tutong
Bandar Seri Begawan
Negara Brunei Darussalam
Tel: (673) (2) 23109 or 23557
TLX: (799) BU2265 RANKOC

Canada
Fluke Electronics Canada Inc.
400 Britannia Rd. East Unit #1
Mississauga, Ontario
L4Z 1X9 Canada
Tel: (416) 890-7600
FAX: (416) 890-6866

Fluke Electronics Canada Inc.
1690 Woodward Drive, Suite 216
Ottawa, Ontario
K2C 3R8 Canada
Tel: (613) 723-9453
FAX: (613) 723-9458

Fluke Electronics Canada Inc.
1255 Trans Canada Highway, Suite 130
Dorval, Quebec
H9P 2V4 Canada
Tel: (514) 685-0022
FAX: (514) 685-0039

Fluke Electronics Canada Inc.
101, 1144 - 29th Ave. N.E.
Calgary, Alberta
T2E 7P1 Canada
Tel: (403) 291-5215
FAX: (403) 291 5219

Chile
Intronica Chile, Ltda.
Casilla 16228
Santiago 9, Chile
Tel: (56) 2-2321886, TLX: (332) 346351

China, Peoples Republic of
Fluke Asia Ltd.
Shun Tak Centre, Suite No. 1501
200 Connaught Road
Central, Hong Kong
Tel: (852) 5-482116
TLX: (780) 87058 Fluke
FAX: (852) 5-479863

Colombia
Sistemas E Instrumentation, Ltda.
Carrera 21, No. 39A - 21, Of. 101
Ap. Aereo 29583, Bogota, Colombia
Tel: (57) 232-4532, TLX: (396) 45787

Denmark
Philips A/S Test & Measurement
Prags Boulevard 80
DK-2300-Kobenhavn S
Tel: (01) 572222, TLX: 31201

Ecuador
Proteco Coasin Cia., Ltda.
P.O. Box 228-A
Ave. 12 de Octubre 2285 y Orellana
Quito, Ecuador
Tel: (593) (2) 529684, TLX: (393) 22085

Proteco Coasin Cia., Ltda.
P.O. Box 9733
Ave. Principal No. 204 y Calle Segunda
Urbanizacion Miraflores
Guayaquil, Ecuador
Tel: (593) (4) 387519

Egypt
Philips Egypt Liaison Office of Philips Export B.V.
10, Abdel Rahman el Rafei Str.
P.O. Box 1687
Dokki, Cairo
Tel: 20-2-3490922
TLX: 228 16 PHEGY UN

Ethiopia
Philips Ethiopia (Priv. Ltd. Co.)
Ras Abebe Areguay Avenue
P.O. Box 2565, Addis Ababa
Tel: 148300

Fiji
Awa New Zealand Ltd.
37 Freestone Walu Bay Road
P.O. Box 858, Suva, Fiji
Tel: (679) 312079
TLX: (792) 2347, FAX: (679) 314379

Finland
Instrumentarium Elektronikka
P.O. Box 64
02631 Espoo 63, Finland
Tel: (358) (0) 5281, TLX: (857) 124426
FAX: (358) 0-502-1073

France
S.A. Philips Industrielle et Commercials
Division Science de Industrie
105 rue de Paris, BP 62
93002 BOBIGNY Cedex
Tel: (1) 49 42 80 80, TLX: 210290

Germany
Philips GmbH
Unternehmensbereich
Elektronik fur Wissenschaft und Industria
Vertriebsbereich Testr & MeBtechnik
MiramstraBe 87 Postfach 310320
D-3500 KASSEL
Tel: (0561) 5010, TLX: 997070

Bereich Fluke Produkts
Oskar-Messter-Strape 18
D-8045 Ismaning

Great Britain
Philips Test & Measurement
Colonial Way
Watford WD2 4TT

Greece
Philips S.A. Hellenique I&E Division
15, 25th March Street
GR-17778 TAVROS or
P.O. Box 3153 Athens
Tel: 30-1-4894911
TLX: 241566-241567

Hong Kong
Schmidt & Co (H.K.), Ltd
18th Floor, Great Eagle Centre
23 Harbour Road
Wanchai, Hong Kong
Tel: (852) (5) 8330-222
TLX: (780) 74766 or 76762
FAX: (852) 5-8918754

India
Hinditron Services Pvt., Ltd.
69/A.L. Jagmohandas Marg
Bombay 400 006, India
Tel: (91) (22) 8121316,
(91) (22) 8125344
TLX: (953) 1175326
FAX: (91) 22-822-0197

Hinditron Services Pvt., Ltd.
8th Main Road
33/44A Raj Mahal Vilas Extension
Bangalore 560 080, India
Tel: (91) 812-363139
TLX: (953) 8452741

Hinditron Services Pvt., Ltd.
5th Floor, "Castle House"
5/1A, Hungerford St.
Calcutta 700 017, India
Tel: (91) 33-432628
TLX: (953) 21-4153

Hinditron Services Pvt., Ltd.
204-5-6 Hemkunt Tower
98 Nehru Place
New Delhi, 110019, India
Tel: (91) 11-6410380
TLX: (953) 316458

Hinditron Services Pvt., Ltd.
Emerald Complex 1-7-264 5th Floor
114 Sarojini Devi Rd.
Secunderabad 500 003, India
Tel: (91) 842 821117
TLX: (953) 04256973

Indonesia
P.T. Lamda Triguna
P.O. Box 6/JATJG, Jakarta 13001
Tel: (62) 21-8195365
TLX: (796) 63938
FAX: (62) 21-819-9631

Iran
Philips Iran Ltd.
(Private Joint Stock Comp.)
P.O. Box 11365-3891, Tehran
Tel: 98-21-674138
TLX: 212545 PHPS IR

Iraq
Philips Medden Oosten B.V.
Baghdad Branch
Hai Al Whida Area No. 902
Street No. 12, Bldg. 141/10
P.O. Box 5749, Baghdad
Tel: 7187181, TLX: 212439 Philips IK

Ireland
P.J. Brennan and Company Ltd.
61 Stillorgan Industrial Park
Stillorgan, Co. Dublin
Tel: 952501, TLX: 3817, FAX: 952333

Israel
R.D.T. Electronics Engineering Ltd.
P.O. Box 43137
Tel Aviv 61430 Israel
Tel: (972) (3) 483211, TLX: 371452
FAX: 972-3-492190

Italy
Philips S.p.A.
Sezione S & I, T & M Dept.
Viale Elvezia 2, 20052 Monza
Tel: (039) 3635240/8/9
TLX: 333343

Japan
John Fluke Mfg. Co., Inc.
Japan Branch
Sumitomo Higashi Shinbashi Bldg.
1-1-11 Hamamatsucho
Minato-ku, Tokyo 105, Japan
Tel: (81) (3) 434-0181
TLX: (781) 2424331
FAX: 81-3-434-0170

John Fluke Mfg. Co., Inc.
Japan Branch
Katsushige Building
2-45 Kohraibashi
Higashi-ku, Osaka 541, Japan
Tel: (81) (6) 229-0871
FAX: 81-6-229-1098

Kenya
Philips (Kenya) Ltd.
OL Kalou Road, Industrial Area
P.O. Box 30554, Nairobi
Tel: 254-2-557999
TLX: 22225 Nairobi

Korea, Republic of
Myoung Corporation
Yeo Eui Do
P.O. Box 14, Seoul, Korea
Tel: (82) (2) 784-9942
TLX: (787) MYOUNG K24283
FAX: (82) 2-784-2387

Kuwait
Al Bahar International Group
P.O. Box 26672 Safat
13127 Safat, Kuwait
Tel: (965) 848601, TLX: (959) 44822

Lebanon
Philips Middle East S.A.R.L.
B.P. 11-670
Ave. Sami Solh, Beyrouth
Tel: 392320/1
TLX: 20765 LE PHIBAN

Malaysia
Mecomb Malaysia Sdn Bhd
P.O. Box 24
46700 Petaling Jaya
Selangor, Malaysia
Tel: (60) 3-7743422, TLX: (784) MA37764
FAX: (60) 3-774 3414

Mexico
Mexicana de Electronica
Industrial S.A. (MEXEL)
Diagonal No. 27
Entre Calle de Eugenia Y Ave.
Colonia del Valle, C.P. 03100, Mexico
Tel: (905) 660-4323
TLX: (383) 1771038
FAX: 525-687-8695

Morocco
Somaciel
304 Boulevard Mohamed V,
Casablanca 05
Tel: 30 80 51/52, TLX: 27021

Nepal
Associated Enterprises
GPO Box 790, Pyaphal Tole
Kathmandu, Nepal
Tel: 13868, TLX: (947) 2568

Netherlands
Philips Nederland
Philips I&E Systems Div.
Building TQIII-4
5600 MD Eindhoven, The Netherlands
Tel: (31) 40-786839
TLX: (844) 35000 PHTC NL
FAX: (31) 40-785968

New Zealand
Philips Scientific & Industrial
Pty Ltd.
2 Wagener Place, P.O. Box 4 021
Auckland 3, New Zealand
Tel: (64) (09) 894-160, TLX: NZ 2395
FAX: (64) (9) 862-728

Nigeria
Associated Electronic Products
(Nigeria) Ltd.
KM 16, Ikorodu Road, Ojota
P.O. Box 1921,Lagos
Tel: 234-1-900160/69, TLX: 21961 NG

Norway
Morgenstierne & Co A/S
Konghellegate 3
P.O. Box 6688, Rodelokka
Oslo 5, Norway
Tel: (47) (2) 356110
TLX: (856) 71719
FAX: (47) 2-381457

Pakistan
International Operations
(PAK Private), Ltd.
505 Muhammadi House
I.I. Chundrigar Road
P.O. Box 5323, Karachi, Pakistan
Tel: (92) (21) 221127, TLX: (952) 24494

Peru
Importaciones y Representaciones
Electronicas S.A.
Avda. Franklin D. Roosevelt 105
Lima 1, Peru
Tel: (51) (14) 28-8650
TLX: (394) 25663

Philippines, Republic of
Spark Radio & Electronics, Inc.
Greenhills 1502, P.O. Box 610
San Juan, Metro Manila,
Philippines
Tel: (63) (2) 775192, TLX: (712) 4005

Portugal
Decada Equipamentos de Electronica
Av. Bombeiros Voluntarios
Lote 102B, Miraflores/Alges
1495 Lisbon, Portugal
Tel: (351) (1) 4103420
TLX: (832) 15515
FAX: (351) 1-410-1844

Singapore, Republic of
Rank O'Connor's PTE Ltd.
O'Connor House
98 Pasir Panjang Road
Singapore 0511
Republic of Singapore
Tel: (65) 4737944, TLX: (786) RS21023

South Africa
South African Philips (Pty) Ltd.
P.O.B. 7703
10 Bond Street, Randburg 2194
Johannesburg 2000
Tel: (27) (011) 889-3911

Spain
Philips Iberica S.A.E. Dpto
Aparatos de Medida
Martinez Villergas 2, Madrid 28027
Tel: 404-2200

Sri Lanka
Computerlink Data Systems, Ltd.
331 Union Place
Colombo, 2, Sri Lanka
Tel: (94) (1) 28641/2, TLX: (954) 22455

Sweden
Philips Foersaeljning AB
Division Industrielektronik
Test & Measurement
S-115 84 Stockholm
Tel: (08) 782 18 00

Switzerland
Philips AG Test-und Messtechnik
Postfach, CH-8027 Zurich
Tel: (01) 4882390

International Sales Offices

Syria
Philips Moyen-Orient S.A.R.L.
Rue Fardoss 79, P.O. Box 2442
Damascus
Tel: 218605/221650
TLX: PHISYR 411203 SY

Taiwan
Schmidt Electronics Corp.
5th Floor, Cathay Min-Sheng
Commercial Building,
344 Min-Sheng East Road
Taipei 104, Taiwan R.O.C.
Tel: (886) (2) 501-3468
TLX: (785) 11111
FAX: (886) 2-5029692

Tanzania
Philips (Tanzania) Ltd.
T.D.F.L. Building - 1st Floor
P.O. Box 20104
Ohio/Upanga Rd.
Dar es Salaam
Tel: 29571/4, TLX: 41016

Thailand
Measuretronix Ltd.
2102/63 Ramkamhaeng Road
Bangkok 10240, Thailand
Tel: (66) (2) 374-2516
TLX: (788) 82796
FAX: (66) 2-374-9965

Tunisia
STIET
32bis rue Ben Ghedahem, Tunis
Tel: 348666, TLX: (934) 13030

Turkey
Turk Philips Ticaret A.S.
Inonu Caddesi 78/80, P.K. 504
80074 Beyoglu-Istanbul
Tel: 90-1-1435910
TLX: 24192 PHTRTR

United Arab Emirates
Philips Middle East
Dubai International Trade Centre
11th Floor
P.O. Box 9269, Dubai
Tel: 971-4-377000
TLX: 46350 PMOA EM

Uruguay
Coasin Uruguaya S.A.
Libertad 2529
Casilla de Correo 1400
Montevideo, Uruguay
Tel: (598) (2) 789015, TLX: (398) UY23010

Venezuela
Coasin C.A.
Calle 9 Con Calle 4, Edif Edinurbi
Apartado de Correos NR-70.136
Los Ruices
Caracas 1070-A, Venezuela
Tel: (58) 2-241-0309, TLX: (395) 21027

Zaire
Philips Electronics S.A.R.L.
137, Boulevard du 30 juin
B.P. 16636, Kinshasa
Tel: 31693, TLX: 21078 KINSHASA

Zambia
Philips Electrical Zambia Ltd.
Mwembeshi Road
P.O. Box 1878, Lusaka
Tel: 218511/218701, TLX: Za 41220

Zimbabwe
Philips Electrical (Pvt.) Ltd.
62 Mutare Road
P.O. Box 994, Harare
Tel: 47211/48031, TLX: 2236

For other countries not listed above:
Fluke Int'l Corp
P.O. Box C9090
Mail Stop 206A
Everett, WA 98206
Tel: 206-356-5511
TLX: 185103 FLUKE UT
FAX: (206) 356-5116

U.S. Fluke Sales Office Area

Alabama

Huntsville
4920 Corporate Drive
Suite J
Huntsville, AL 35805-6202
(205) 837-0581

Arizona

Tempe
2211 S. 48th Street
Suite B
Tempe, AZ 85282
(602) 438-8314

Tucson
(602) 790-9881

California

Northern
46610 Landing Parkway
Fremont, CA 94538
(415) 651-5112

Irvine
P.O. Box 19676
Irvine, CA 92713-9676
16969 Von Karman
Suite 100
Irvine, CA 92714
(714) 863-9031

Burbank
(213) 849-7181

Los Angeles
(213) 747-5935

San Diego
(619) 292-7657

Colorado

Denver
14180 E. Evans Ave.
Aurora, CO 80014
(303) 695-1000

Connecticut

Hartford
Glen Lochen East
41-C New London Turnpike
Glastonbury, CT 06033
(203) 659-3541

Florida

Clearwater
(813) 799-0087
Miami
(305) 462-1380

Orlando
940 N. Fern Creek Ave.
Orlando, FL 32803
(407) 896-4881

Tampa
(813) 251-9211

Georgia

Atlanta
2700 Delk Road
Suite 150
Marietta, GA 30067
(404) 953-4747

Illinois

Chicago
1150 W. Euclid Avenue
Palatine, IL 60067
(312) 705-0500

Indiana

Indianapolis
8777 Purdue Road
Suite 101
Indianapolis, IN 46268
(317) 875-7870

Louisiana

New Orleans
(504) 455-0814

Massachusetts

Boston
Middlesex Technology Center
900 Middlesex Turnpike
Building 8
Billerica, MA 01821
(508) 663-2400

Maryland

Baltimore
(301) 792-7060

Rockville
5640 Fishers Lane
Rockville, MD 20852
(301) 770-1570

Michigan

Detroit
45550 Helm Street
Metro West Industrial Park
Plymouth, MI 48170
(313) 522-9140

Minnesota

Eagen
1380 Corporate Center Curve
Suite 113
Eagan, MN 55121
(612) 854-5526

Missouri

St. Louis
11756 Borman Drive
Suite 160
St. Louis, MO 63146
(314) 993-3805

North Carolina

Greensboro
1310 Beaman Place
Greensboro, NC 27408
(919) 273-1918

New Jersey

Paramus
P.O. Box 930
Paramus, NJ 07653-0930
West 75 Century Road
Paramus, NJ 07652
(201) 262-9550

New Mexico

Albuquerque
(505) 881-3550

New York

Rochester
4515 Culver Road
Rochester, NY 14622
(716) 323-1400

Ohio

Cleveland
Plaza South Three
Suite 402
7271 Engle Road
Middleburg Heights, OH 44130
(216) 234-4540

Oklahoma

Northeast
(405) 236-2977

Tulsa
(918) 665-3530

Oregon

Portland
(503) 227-2042

Pennsylvania

Malvern
200 Lindenwood Drive
Malvern, PA 19355
(215) 647-9550

Pittsburgh
(412) 261-5171

Texas

Austin
(512) 459-3344

Dallas
1801 Royal Lane
Suite 307
Dallas, TX 75229
(214) 869-0311

El Paso
(915) 533-3508

Houston
(713) 240-5995

San Antonio
10417 Gulfdale
San Antonio, TX 78216
(512) 340-0498

Utah

Salt Lake City
(801) 268-9331

Washington

Seattle
5020 148th Ave. N.E.
Suite 110
Redmond, WA 98052
(206) 881-6966

Washington, DC

Washington, DC
(301) 770-1570

Appendix H
Known Bugs in Version 4.0 Software

The following is a list of known bugs in version 4.0 software with descriptions and workarounds:

Bug Number: 870708-08: **TL/1 sometimes loads programs more than once.**

DESCRIPTION

Occasionally unusual disk activity is observed when a TL/1 program is executed in a loop. These programs are usually pod special programs.

The problem is that the program is named FOO but is invoked as foo. The operating system looks for the program on disk case insensitively, and when it finds the program, it executes it. However, the operating system stored the program in a case sensitive symbol table, as required by the published semantics of TL/1. The next time the program is invoked, the symbol table is searched for foo. FOO is in the symbol table, but not foo, so the program is loaded again.

The bug is that the actual name of the program was not checked once the program was loaded.

WORKAROUND

The workaround for this bug is to correctly invoke the program FOO as FOO. The name of a program must be spelled the same everywhere, and this spelling must match the name in the program statement.

No correct programs are broken.

This bug is left unfixed so that incorrect programs that 'seem' to work can continue to be used for awhile.

Bug Number: 880115-26: **TL/1 should preserve linkage and space when exercising faults.**

DESCRIPTION

Space and linage should be the same for each iteration of an exerciser. It isn't being newly set, so if an exerciser does a setspace(), bad things can happen.

Bug Number 880122-08: **TL/1 Debugger can run out of stack space.**

DESCRIPTION

When repeatedly running a program from debugger looking for random error. Got "Infinite Recursion or Stack Overflow"

NOTE

Program always generates a fault. Repeated by:

F10 (Fault)
F4 (Execute)
Enter

This is caused because original execution is just stalled, and new execution starts a new job. If execute is the main program instead of an internal function, it should flush the old program.

WORKAROUND

The INIT key discards all nested program invocations, clears all open file channels, reclaims RAM, and restarts TL/1's run time environment. Doing this solves the problem for now, with the only side effect being that the breakpoints are erased.

Bug Number 880323-02: **Intermittent 9105 error when rebooting.**

DESCRIPTION

1. Reset 9105A without USERDISK.
2. Install USERDISK in DR2.
3. Hit Reset.
4. Message - Podname does not match.
5. Hit Reset.
6. All OK.

Correct Pod Database is picked up second time not first.

Additional Info - if any disk is in DR1 the problem goes away. Also DR1 light goes on if disk is in the drive.

WORKAROUND

Have a disk in Drive 1 when you reboot on 9105.

Bug Number 880401-02: **Marking characters on long line can mess up attributes.**

DESCRIPTION

Bug involving MARKing characters on the bottom line of the screen so that the screen has to scroll. Line attributes (bolding) are wrong. To reproduce:

1. Edit a textfile. Put a long line in it that takes up the rows on the CRT. Position it on the CRT so that the first row of the long line is on the bottom of the CRT. (The second row of the long line is off the screen.) Put the cursor on the first row of the long line.

2. Press MARK and right arrow until the screen scrolls and you have marked a few characters on the second row of the long line.

3. Press left arrow a few times. The bolding is not removed as the cursor backs up.

NOTE

The editor does YANK/CUT the correct number of characters, it is only the bolding that is incorrect.

Bug Number 880425-02: **editor positions cursor improperly on long lines.**

DESCRIPTION

Cursor improperly positioned - Press <End Line> key on long line* or Press <Begin Line> and then Return - note changes example below:

```
 declare
   global string dvm = " 02" !8840 on IEEE address 2
   global numeric ieee_out
   global numeric ieee_in
   global numeric disp
 end declare

* ieee_out = open device "/port1", as "output",
    mode "unbuffered" !CR print term
  ieee_in = open device "/port1", as "input"
  disp = open device "/term2", as "output"
```

Bug Number 880808-04: **Bogus error message when out of memory in TL/1.**

DESCRIPTION

BOGUS ERROR MESSAGE WHEN OUT OF MEMORY.

I executed a large TL/1 program from the front panel (2 megabyte hard model) and repeated it several times. At seemingly random times it would quit with a “NO DISK IN DRIVE” error message.

*I just tried to repeat this without success. But this time the system behaved as though I had pressed “LOOP” because every time I exited the program, it would restart!

I know this has something to do with running out of memory because there is not enough memory available for OS-9 to even run “mfree.” I get a “NO RAM AVAILABLE” error message when I try.

Bug Number 880815-01: **Deficiencies in DEFINE/TEST/SHOW RAM REF.**

DESCRIPTION

DEFINE/TEST/SHOW RAM REF

Has problem with ADDR option. SHOW does not show which option is used. TEST does not allow the ADDR option to be shown or changed.

TEST sometimes comes up with illegal ADDR step message.

Bug Number 880909-04: **Application interface loses cursor.**

DESCRIPTION

1. Place the front panel cursor in rightmost field of the address option display (DWORD of 80386 MEMORY DWORD).

2. Execute a program that leaves the machine in a memory space with less fields in the address option. (I/O space of 80386.)

3. Try to do a read or other function from the front panel. When you go down to the address option field, the cursor disappears.

WORKAROUND

A left arrow key allows the cursor to appear.

Bug Number 881005-04: **TL/1 has problems loading programs containing control codes.**

DESCRIPTION

A program which loads into the editor properly, passes the F10 "CHECK" function, and executes from the front panel causes the following message,when the F3 "DEBUG" key is pressed:

```
Program '$debug$' could not be loaded properly.
```

Various versions of this program have run through the debugger earlier.

WORKAROUND

The program in question is written off-line and contained strings with control character codes that the 9100A editor does not insert. These one character codes were expanded into three character escape sequences that caused the end of the string to overwrite the next couple of characters. Although the program on disk had no syntax errors, the program saved in the temporary file of the debugger did have syntax errors, and the debugger refused to load it.

Programs edited off-line must contain only printable characters.

Bug Number 881007-01: **Replacing Single Quote With Double Quote.**

DESCRIPTION

On the following TL/1 line:

t = ‘t’

I did a REPLACE ‘ with “ and it worked for the first quote. When I did a SHIFT REPLACE for the second quote, the following error message is displayed:

```
String not closed by end of line. Missing double
 quote?
```

I was expecting the replacement operation to occur.

Bug Number 881013-05: **Relative File Names Don’t Work if UUT is Unspecified.**

DESCRIPTION

If a TL/1 program is executed without specifying a UUT name (for example, a program in the pod library found under the POD key), or if I try to open or delete file “foo”, the program exits with a fatal error message. I expected it to default to /HDR/FOO.

This is only a problem when a UUT is not specified on the front panel, and when debugging in the podlib or proglib.

WORKAROUND

Make sure that a UUT is specified before the program is executed.

Debug podlib and proglib programs in a UUT.

Bug Number 881019-04: **DEFINE/TEST RAM REF size problems.**

DESCRIPTION

Define multiple RAM references, some with DWORD size and addr increment of four, some with WORD size and increment of two, some with BYTE size and increment of one. (Can even be same space.)

TEST RAM FAST ALL REF will return error of illegal step size.

Bug Number 881020-02: **operator's i/f doesn't clear prompts.**

DESCRIPTION

Old prompts remain when new appl key is pressed. How to: Do a TEST BUS with 9132, (=PASSED appears). Press the RAM key and select HYPER TEST (=PASSED may remain?) then ENTER. The HYPER test prompts appear, then press the BUS key. The TEST BUS appears on the left with old Hyper stuff on the right.

Bug Number 881020-03: **Key sequences execute unintended previous function.**

DESCRIPTION

Key Sequence Funny? Press the PROBE key and softkey to select the FREQ softkey then, ENTER. Press SEQ BEGIN ENTER. Press PROBE ARM ENTER. Press SEQ QUIT ENTER. Press SEQ PERFORM ENTER. It does a FREQ measurement?

Bug Number 881020-04: **Setup fields don't have INC and DEC**

DESCRIPTION

The numeric fields under SETUP such as TIMEOUT and INTERFACE XFER_ADDR do not have the INC and DEC softkeys?

Bug Number 881025-01: **Debugger sometimes cuts off the end of error messages.**

DESCRIPTION

While testing the Demo/Trainer software on version 3.85, I came across a bug in the Demo program which put the message to the Application Interface: Variable 'contention_checked' used in expression before set."

I ran the same programs from the debugger to get a clue on which program had the problem. When the error window opened the following message was displayed: "In 'tst_conten' line 103 - Variable 'contention_checked used in expression be." The error message was truncated to 76 columns.

Bug Number 881028-01: **TEST RAM FULL [ALL] REF doesn't always allow the ADDR option.**

DESCRIPTION

TEST RAM FULL REF or ALL REF do not always allow the ADDR OPTION to be selected? They are stored, I thought, then they should not be selectable...right? If they are stored and not selectable, then try DEFINE for MEMORY BYTE ENTER and then TEST down-arrow to change ADDR OPTION: then ENTER (changes when complete to ?last define one?).

Bug Number 881028-03: **Bad display of READ DATA**

DESCRIPTION

If reading a virtual address with an extended address and address are of a length such that a single bit of read data would be displayed at the very end of the screen, the data is NOT displayed, and no more info led comes on.

Example: Read Virtual with 7-bit extended address and 5-bit address returning data if 9.

Bug Number 881028-08: **TL/1 program suddenly runs very slowly.**

DESCRIPTION

A TL/1 program that ran under previous releases ran extremely slow under 3.85 software.

WORKAROUND

The bug is pathological case in TL/1's memory manager. There is so little memory free that TL/1 spends a lot of time scavenging, but just enough that TL/1 doesn't have to ask the operation system for more.

To make this behavior go away, declare an array (start with 50 elements, and add 50 elements at a time until the problem disappears). Sometimes a couple of comment lines in the program will also get rid of the bug.

This bug will be repaired in release 4.1.

Bug Number 881031-02: **Disk ID error in copy.**

DESCRIPTION

Using a 9100, I formatted a disk from appl and then did a COPY /DR1 to /DR1.

The copying was done in a single pass. Then I tried to edit /DR1 type USERDISK and got the following message:

```
Disk ID error
```

I can eventually edit /dr1 by RESETing the system twice.

WORKAROUND

RESET the system a couple of times.

Index