HP 82000 IC Evaluation System

Using the HP 82000

Models D50, D100, D200 and D400

SERIAL NUMBERS

This manual affects all systems with software revisions up to 2.0.0

HEWLETT PACKARD

Legal Information

Notice

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

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MANUAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Printing History

New editions of this manual will incorporate all material updates since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

October 1990

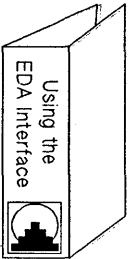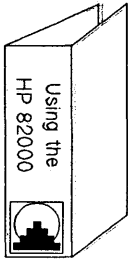**Revision 1.0 (Software versions up to 2.0.0)**
This manual, together with the manual *Standard Test Function Reference* (part no. E1280-90205), replaces the earlier manuals *Getting Started with the HP 82000* (part no. E1280-90201) and *Using the HP 82000* (part no. E1280-90202).
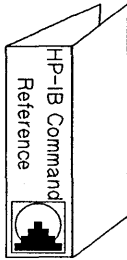
Documentation Map

The following figure shows the User manuals available for the HP 82000 IC Evaluation System.

HP 82000 USER DOCUMENTATION

Getting Started   Reference   Advanced

Using the HP 82000

Standard Test Function Reference

Advanced Testing with the HP 82000

Test Function Programming Manual

Using the EDA Interface

HP-IB Command Reference

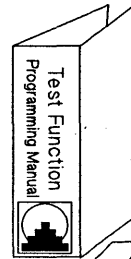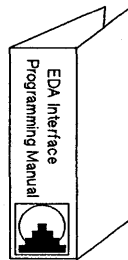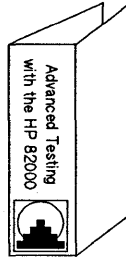EDA Interface Programming Manual

High Speed Testing Introduction and Reference Guide
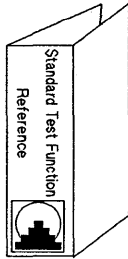
Installation and Service Documentation:

Site Preparation and Planning Guide

Installing the HP 82000

Maintaining the HP 82000

Servicing the HP 82000

Preface

Purpose

The purpose of this manual is to describe the use of the HP 82000 IC Evaluation System, and to give you a basic understanding of the features and operation of the system.

chapter 1   gives an introduction to IC testing with the HP 82000, and briefly describes the operating principles and architecture of the system;

chapter 2   explains the basic procedures involved in wiring your device to the DUT board;

chapter 3   explains how to configure the HP 82000 to the pins of the device under test;

chapter 4   explains how to set up the logic levels for your device;

chapter 5   explains how to enter timing data to the HP 82000;

chapter 6   explain how to set up the vector data needed to test the device;

chapter 7   explains how to use the prewritten test functions to test your device using the interactive software screens;

chapters 8, 9, and 10   explain how to use and interpret the test result displays;

Appendix A   lists the Test Function error codes;

Appendix B   describes the automatic timing correction feature.

Contents

7. Using the Standard Test Functions

1

Introduction to the HP 82000

This chapter describes the fundamental aspects of IC testing with the HP 82000 and gives an overview of the system architecture.

Starting Up the System

From the HP-UX prompt on your Programming Workstation (PWS), you can start the HP 82000 system software by typing in:

hp82000 -*options*

The command line options are:

-o   **offline:** to start the software without accessing the hardware.

-i   **initialize PWS from hardware:** to retrieve the current hardware states, and display them in the setup windows.

-x   **X Windows:** to start the system software while the workstation is already running X Windows.

-b   **background:** to start the software from X Windows, but still allow access to the windows already running on your workstation.

First-Time Users   Each time you use the system, a file .device_inuse is
stored in your home directory, and contains the path to
the last device that you used during your last session.

If you have never used the system before, this file does
not exist and the system asks you which device you wish
to test.

When you start up the system, the main menu is
displayed in gray-scale (as usual) along with the
Welcome Window. The Welcome Window is then
overlaid with the dialog box shown in Figure 1-1.

```
┌───────────────────────────────────────┐
║                                         ║
║   create new or enter existing device   ║
║                                         ║
║                         ( create )      ║
║                                         ║
║                         ( enter  )      ║
║                                         ║
║                         ( cancel )      ║
║                                         ║
└───────────────────────────────────────┘
```
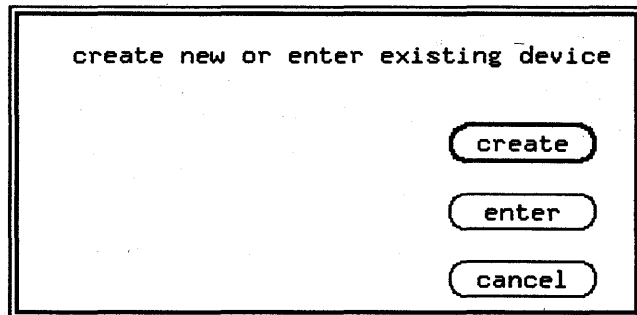
Figure 1-1. First-time User's Dialog Box

What you now do depends on whether you are testing
a device that has not been tested on the system before,
or whether you are developing a new test for an already
existing device.

If you create a new device, the system will create
directories and files for the device as shown in Figure 1-3
and Figure 1-2.

```
                        <Destination_Directory>
                                 |
                           <Device_Name>
  ┌──────┬───────────────┬────────┬────────┬───────────┬─────────┐
  |      |               |        |        |           |         |
calib.raw configuration  levels   report   test_func   vectors

  calibration      format  pin_attributes  test_flow    timing      waste
```
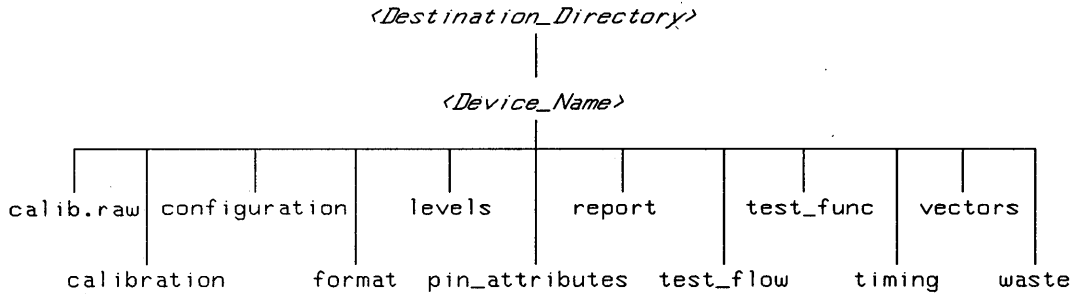
Figure 1-2. Device Directories for Setup and Test Functions

```
                        <Destination_Directory>
                                 |
                           <Device_Name>
        ┌────────────────┬────────────────┬──────────────┐
        |                |                |              |
   cae_control      cae_params        cae_report

        cae_data              cae_pmap
```
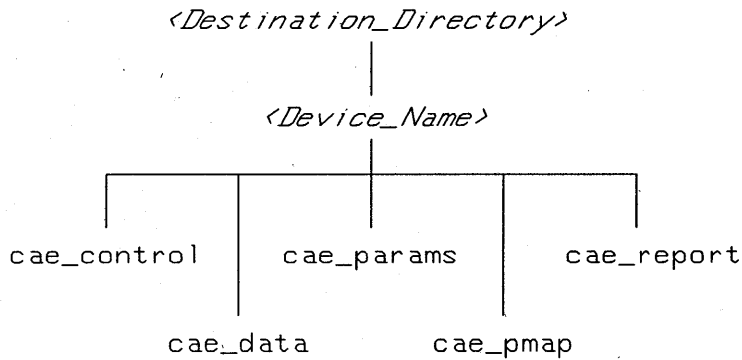
Figure 1-3. Device Directories for the EDA Interface

Creating a New Device

If you want to create a new device, click the ⬡ create ⬡ pushbutton. The **create new device** dialog box is then displayed, similar to the one shown in Figure 1-1. Before the system will create a new device, you need to fill in the following edit boxes:

path name   is the path to the destination directory where the device will be created (refer to Figure 1-2 and Figure 1-3).

device name	is the name of the device. You can enter any name that you want but it makes sense to choose one that you will recognize later.
device technology	tells the system what type of technology the device uses (e.g ECL, CMOS, etc..). Depending on the device technology, the system assigns the correct logic-level default values for the standard test functions, and the correct calibration files.

Note 👆

There are two technology setups supplied with the system; ECL and CMOS. If your device is not fabricated in one of these technologies, refer to the manual *Advanced Testing with the HP 82000* for information on how to create new device technology setups.

Click the (create) button to create the new device. The system checks that the parameters entered are correct, and then creates the directory structures shown in Figure 1-2 and Figure 1-3.

Setup and Test Directories

The following directories are used by the setup and test software to store the data that you enter in the setup windows. The manual *Maintaining the HP 82000* gives further details about the calibration file hierarchy.

calib.raw	contains "raw" calibration measurements generated by the calibration utility. This data is then processed to create the final calibration offset values for the device under test.
calibration	final calibration offset data for the current device.

configuration   the pin configuration that is derived
                    either from the EDA interface,
                    or one that you enter in the Pin
                    Configuration window.

format   the different display formats for
                    each window in the Setup and Test
                    software.

levels   the level setups that you enter in the
                    Level Setup Window.

pin_attributes   contains timing adjustment values
                    for each pin-driver and receiver, and
                    the values of series resistors (R_s)
                    wired onto the DUT board (used for
                    resistive divider arrangements). The
                    fine-corrections to the Standard AC
                    Calibration data allow you to tune
                    the system for special applications.
                    These values are displayed in the Pin
                    Attributes Setup window. For most
                    applications, the pin_attributes
                    file remains empty.

                    The manual *Advanced Testing with
                    the HP 82000* gives full details about
                    applications for using pin attribute
                    files.

report   contains the report files generated by
                    the system.

testflow   all the testflows that you write
                    for the current device, using C or
                    BASIC.

testfunc   all the functional tests that you have
                    setup and saved for the device.

timing   the timing setups for each pin in
                    the system. These are derived by

the EDA interface, but can also be
entered by the user.

user_calibration calibration files generated by
performing an AC Calibration at
user settings.

vectors the vectors for functional testing,
and the tests set up in the Sequencer
Control and Sequencer Programming
Windows.

waste the default directory when you enter
the HP-UX shell.

EDA Directories

The following directories are used by the Electronic
Design Automation Interface. For more information
about the contents of these directories, refer to the
manual *Using the EDA Interface.*

cae_control state mapping files. Map EDA states to
allowed HP 82000 states.

cae_data EDA files that the EDA Interface
translates.

cae_params contains setup parameters for the EDA
Interface windows.

cae_pmap pin mapping information. Maps EDA
pin names to HP 82000 pin names.

cae_report contains status files about the EDA
translation.

After creating this directory structure, the system
informs you that no calibration data exists for this
device. You can either copy across existing calibration
data for a similar device, or you can carry out the
Standard AC Calibration routine. The manual

Maintaining the HP 82000 describes how to performing calibration routines.

Once the new device has been created, you can use the EDA interface to process your CAE files, and produce the necessary setup and vector files for testing (refer to the manual *Using the EDA Interface*).

Note

The EDA Interface cannot set up the high and low logic-levels; you *must* set these levels yourself in the Level Setup Window before you start testing.

Accessing an Existing Device

If you are a new user but you want to work on an existing device, click the (enter) button. The enter device dialog box is then displayed similar to Figure 1-4.

path name

/users/demo

enter device

enter device

cancel
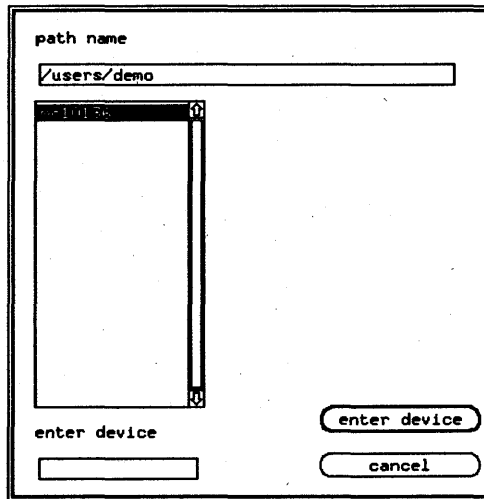
Figure 1-4. Accessing an Existing Device

You need to enter the path to the device in the pathname box. A list of all sub-directories for that path

will be displayed in the browser. You can either use the mouse to select one of these directories (devices), or you can enter the device name in the edit box provided.

Once you have made your selection, click the (enter) select button. The system will then load the necessary setup files for that device and you can begin testing.

Using the Setup Windows

The Setup windows are selected from the double row of pushbuttons which form the main menu (across the top of your PWS screen). The Setup windows show the current settings of the hardware, and you can view these settings and edit them using the appropriate setup window.

With most of the Setup windows, any values that you edit immediately take effect in hardware. The exception is the Pin Configuration window; you must click the (download) button to enter these Setup values into the hardware. (Because all the other windows are subordinate to Pin Configuration, this safety feature prevents accidental erasure, or re-configuration, of your Setup data).

The device setup windows are:

Pin Configuration window — Maps the tester resources to the pins on your DUT. Once mapped, the tester channels are represented by the pin names (for example, "clock", for the clock input pin, "Qn" for output pins, etc ...). You must complete the Pin Configuration setup before you use any of the other setup window. Changes to the three other setup windows always refer to the pin names, not to the tester channels.

Level Setup window	Sets: the logic levels for the drive signals; the threshold levels for the received signals: and the dc levels of the device power supplies.
Timing Setup window	Sets: the edge placement for the drive signals; the timing for comparisons in Edge Compare and Window Compare modes; and the format of the drive signals (e.g. DNRZ, RZ, RC, etc..).
Vector Setup window	Allows you to edit the vector data generated by the EDA Interface, or to input patterns of vector bits using the built-in pattern generator. Controls functional testing and provides access to sequencer programming.

As well as the data that you enter in these four main Setup windows, for particular applications the tester may need additional information about the way you have wired up the DUT board for your device.

Pin Attributes Setup window	This window allows you to set: the values of series resistors wired on the DUT board (in a resistive divider arrangement); individual timing offsets for the driver (input) and receiver (output) edges of each I/O channel (e.g. to allow for additional wire lengths on the DUT board); and an adjustment to the system roundtrip (e.g. to temporarily adjust the driver to receiver delay for a special application). In most cases, the fields in this window will be left empty.
	The manual *Advanced Testing with the HP 82000* gives full details about applications for using pin attribute files.

For a complete tester setup, the following data has to be downloaded from the PWS to the hardware:

1. Pin Configuration

2. Other Setup Parameters

    a. For each DUT input pin (HP 82000 driver channels):

      i. timing and signal format

      ii. levels

      iii. test data (vectors)

    b. For each DUT output pins (HP 82000 receiver channels):

      i. timing

      ii. levels

      iii. expected data (vectors)

The Pin Configuration, Timing, Level and Vector Setup data is loaded from the separate HP-UX subdirectories of your device directory, described in "Setup and Test Directories." Each of these subdirectories contain files that can be downloaded to the hardware. These files can be created, edited, deleted and downloaded from the software windows (by using the `File` pulldown menus).

Using LAST_SETTING files

Often, each of the Setup subdirectories contains a file called LAST_SETTING. These files contains the hardware settings when the system software was last stopped (using the (QUIT) button). When you restart the system software, the Setup values in these files are used in preference to any other Setup files.

You can inhibit the loading of LAST_SETTING files by starting the software with the -i option on the HP-UX command line. This causes the current hardware settings to be read, and allows you to recover the tester settings after a system failure, software crash or accidental quit without saving the tester state.

If the LAST_SETTING file for the Pin Configuration exists, it is downloaded to the hardware first. If the Pin Configuration loads successfully, then the LAST_SETTING files for Vector, Level and Timing Setups are loaded.

If the LAST_SETTING file for Pin Configuration does not exist or does not download successfully, a warning message in the Report Window informs you of this. If a LAST_SETTING file for any of the other setups is missing, the hardware will load the power-up default values from the firmware.

Note

As well as LAST_SETTING files for Setup data, there is also a calibration LAST_SETTING file. The manual *Maintaining the HP 82000* gives full details of the interrelationship between calibration files.

Sequence of Loading Setup Files

Figure 1-5 shows the procedure that the software follows, during start-up, when it decides which setup files to use.
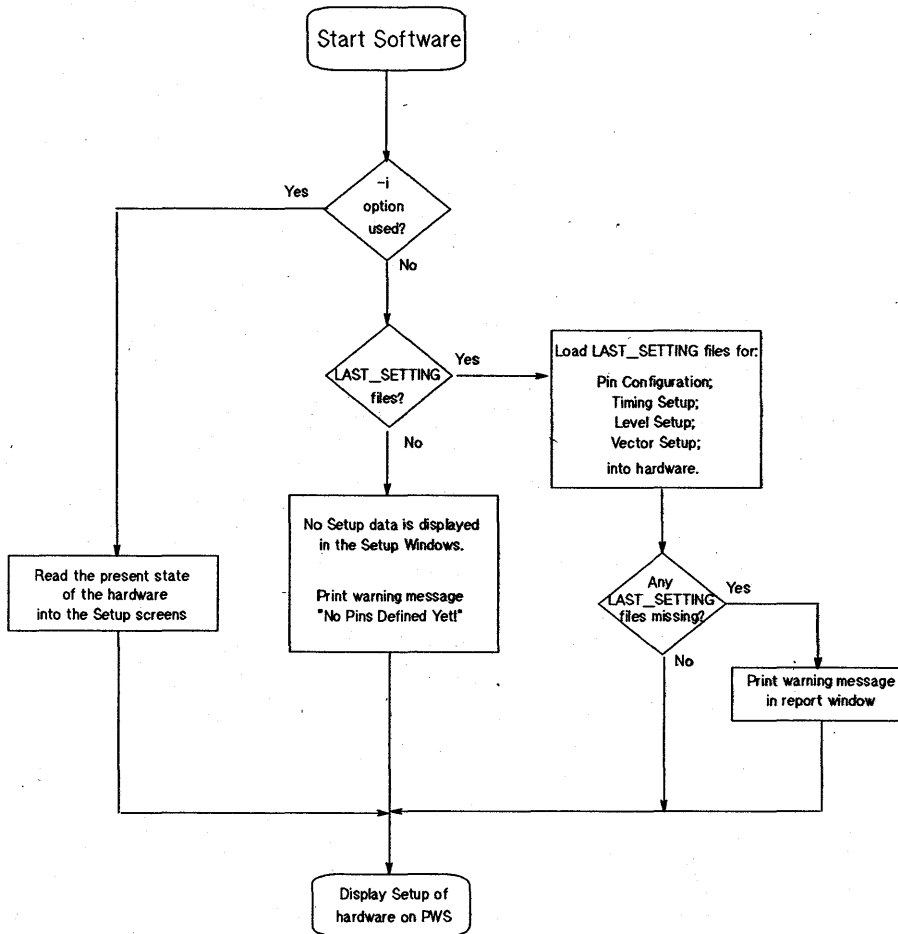
Figure 1-5. The Setup File Loading Sequence

Storing Last Used Values on Quitting

When you want to quit the HP 82000 System Software, you will be given a chance to save the last Setup values that you used (that is, the settings currently stored in hardware), in the form of LAST_SETTING files.

The System Administrator dialog box is presented in the center of the screen, when you click the (QUIT)

pushbutton - shown in Figure 1-6. The default is *quit without saving changes:* the store 'LAST_SETTING' checkbox is empty.

If you now quit, no LAST_SETTING files are created.
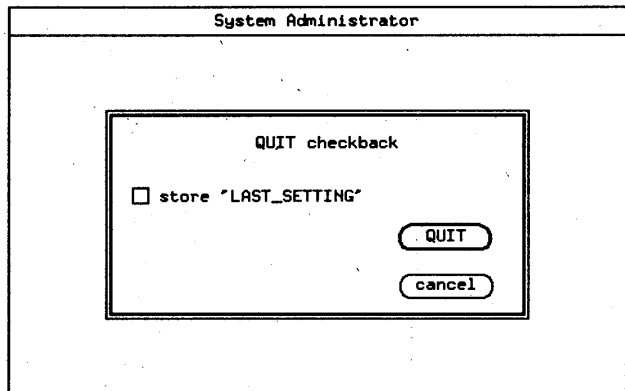
Figure 1-6.
System Administrator Dialog Box - Quitting System Software

If you click the checkbox and quit, LAST_SETTING files for:

- Pin Configuration;
- Level Setup;
- Timing Setup;
- Vector Setup;
- Pin Attributes Setup;
- and Calibration

are created. You can now power down the tester hardware without losing the last values that you used.

Editing in the Window Environment

Any entries or modifications in a setup window are made in text edit fields. Before an entry in a setup window can be edited, its edit field must be activated. An edit field that has been activated has a rectangular border drawn around it. This is called an entry field. To activate a field, move the mouse pointer to it and click the left mouse button. Once a field is activated, you can move the entry field around the Setup Window using the keyboard:

To move left   press the (Shift) (Tab) key

To move right   press the (Tab) or (Return) keys

To move up   press the (▲) key

To move down   press the (▼) key

You can also use the mouse to move the entry field around the window.

In addition, a text cursor appears inside the entry field. It pinpoints the character that can currently be edited inside the entry field. You can move the text cursor left and right using the mouse or using the keyboard as follows:

To move left   press the (◀) key

To move right   press the (▶) key

When you deactivate a text edit field and activate another one, the changes made within it are downloaded to the hardware, but see below. There are certain differences in operation between the Pin Configuration Window and the Timing, Level and Vector Setup Windows.

Pin Configuration Window

The Pin Configuration window has two modes of operation:

1. the monitor mode

2. the editor mode

After activating the Pin Configuration Window, the monitor mode will come up, just displaying the actual hardware settings. In this mode you can scroll the display and read the loaded values. The display shows the actual status of the hardware.

To change any of the entries in the window you must first enter the editor mode. You can now make further changes or additions to the Setup Pins configuration. You can also scroll and read the current settings.

When you have made all your entries in the Pin Configuration window, you must download the new configuration to the hardware by clicking the (Download) pushbutton.

If you have made any errors in the Pin Configuration, (such as assigning the same channel twice) the system software responds with appropriate error messages at the time of downloading your configuration to the hardware. Further details are given in Chapter 3.

Timing, Levels and Vectors Windows

The settings that you enter in the Timing Setup, Level Setup and Vector Setup windows are downloaded to the tester hardware as soon as you leave an entry field. Thus the setup values that appear in the windows are always the actual values present in the tester.

A download to hardware occurs for example by closing the Setup Pins editor window. Any error messages will be displayed at the point of downloading the settings. The full details are covered in the relevant Setup chapter.

Programming Setup Parameters Offline

You can still program your setup parameters even if no tester hardware is connected to your workstation. To start the system software in offline mode, at the HP-UX prompt type:

hp82000 -o

In the offline mode you can use and program all setup windows, develop test programs and work with vector files. You cannot run any tests or call up any of the results windows.

When programming the setup windows offline, the system is not capable of performing error checking, such as voltage level or context dependencies, which depends on the presence of the hardware. Full error checking takes place only when the setups are downloaded to the hardware.

Simulating Mixed D200/D400 Configurations Offline

If your D400 tester has any 200 MHz I/O boards fitted, they are automatically detected by the software. The additional fill-menu options for 200 MHz boards are then available in the setup windows.

However, when you use the workstation for offline programming of a D400 system, you must enter details of the I/O boards that are in the hardware. The file for simulating the tester configuration has the HP-UX pathname:

/hp82000/pws/data/offl_tester_co

Each line of this text-file represents one mainframe. The sixteen slots of the mainframe are each given a number— to represent a 200 MHz board, a 400 MHz board, a PMU or a blank slot. The numbers are separated by a comma.

The complete syntax for each mainframe looks like this:

slot 1 , slot 2 , slot 3 , ... , slot 16

where each slot is either:

400	—representing a 400 MHz board
200	—representing a 200 MHz board
100	—representing a 100 MHz board
50	—representing a 50 MHz board
1	—representing a PMU board
0	—representing an empty slot

For example:

400,400,400,400,400,400,400,400,1,0,0,0,0,0,0,0
400,400,400,400,400,400,200,200,1,0,0,0,0,0,0,0

This indicates a two-mainframe system, with :

- 200 MHz boards in slots slots 7 and 8 of the 2nd mainframe;

- 400 MHz boards in slots 1, 2, 3, 4, 5, 6, 7 and 8 of the 1st mainframe, and slots 1, 2, 3, 4, 5 and 6 of the 2nd mainframe;

- PMUs in slot 9 of each mainframe;

- and all other slots left empty.

Performing Tests

All testing starts with the Test Control Window, which you access by pressing the (Test Control) button on the main system menu. The Test Control Window is displayed as shown in Figure 1-7.

This manual describes how to use the built-in test functions interactively. You can find detailed descriptions of the tests themselves, and information about the input parameters of the test functions, in the manual *Standard Test Function Reference.* Information about linking tests together into programmed test flows

is covered in *Advanced Testing with the HP 82000*. A
summary of the test functions available is given in
Chapter 7.
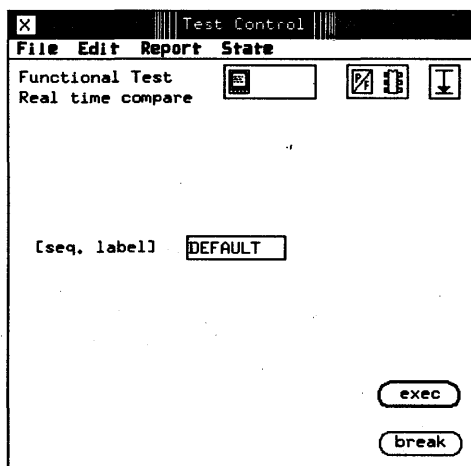
Figure 1-7. The Test Control Window

Choosing the Test

From the library of standard test functions, you can
choose the appropriate test for your device using a series
of menus. The parameters that you have defined in the
Setup windows form the test data; the test function
downloads this data to the DUT, evaluates the DUT
outputs, and collates the results.

Accessing the Test Functions from the Test Control Window

All of the test functions and control functions can be
accessed via the new option in the file pulldown menu.

The default group is AC Tests. You can select the test
group you want via a fill menu in the group entry field
at the top of the window. The test functions available in
the group you choose, are listed in the browser.

Saving Test Function Settings

When you have set up parameters for a particular test on your DUT, and you want to save the parameters for use later, you can do this by using the **Save** option in the **File** pulldown menu. These filled-in tests are saved in the directory `../testfunc`.

Recalling Test Functions

You can recall these filled-in tests via the **Load** function of the **File** pulldown menu or use them in BASIC or C test flows.

Modes for Testing

There are three modes that can be set when using the standard test functions:

- Report Mode - to select how the test results are displayed
- Test Mode - to select whether the tests return only pass/fail information or measure a value, and whether tests are performed on single pins or on all pins at the same time
- Repeat Mode - to select how many times the test function is to be executed

The current status of the modes for testing is indicated by icons in the top right hand corner of the Test Control window.

2

Wiring the DUT Board

To test your device with the HP 82000, you must wire it into a special printed circuit board; this is the *DUT board.* Before you can start setting up your test parameters, you must decide which channels of the HP 82000 are going to be electrically connected to which pins of your DUT, and then wire your DUT board accordingly.

The following sections give you general details about connecting up your DUT board. You can find specific information about the positions of the pads on your DUT board, either in the literature supplied with each board, or on the silk-screen printed legend on the board itself.

Selecting a DUT Board

You must consider the following areas when you select a DUT Board:

Tester Hardware   DUT boards are available either as full sized boards (for Standardframe systems or Maxiframe systems) or as half sized boards (for use with all systems).

No. of Channels   This will determine which size of board you need.

| Impedance | 100Ω for the D50 System |
| | 50Ω for the D100, D200 and D400 Systems |

Wiring	**Standard Boards.** You must make all the wire connections yourself.
	Pre-wired Boards. These have printed wiring tracks for either DIL (Dual in Line) or PGA (Pin Grid Array) packages.
	Customized Boards. Services are available for making boards with special connection sockets, channel connections and with special components installed.

Purpose	**Packaged IC:** Grid array for DUT test-fixture.
	Prober Interface: cabling connection to probe card.
	General Purpose: for cable connection of: accessories, modules, handler.

The *HP 82000 Ordering and Configuration Guide* has full details about the wiring options available with Pre-wired DUT boards.

DUT Interface

The DUT board fits into an interface assembly on the front of the tester. The DUT interface comprises two main mechanical parts:

■ **Bottom Frame** - housing blocks of pogo-pins, carrying DUT signals. Fitted to the bottom frame are 4 mounting posts which locate the DUT board inside the interface.(Figure 2-1)

■ **Top Frame** - holding the DUT board pads firmly against the pogo pins, ensuring a good electrical contact. The top frame is hinged at one end, with two clamp handles at the other end.
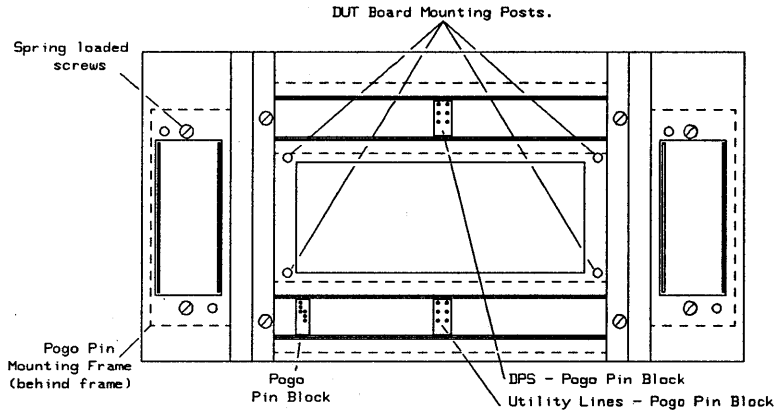
Figure 2-1. Bottom Frame Showing Mounting Posts
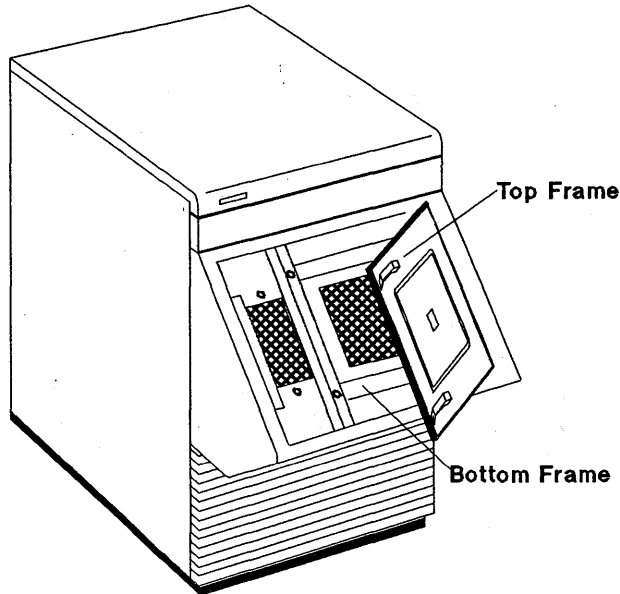
Figure 2-2. The DUT Interface of the HP 82000 Standardframe

Pogo Pin Layout

The pogo pins are spring-loaded, gold-plated contacts, which are mounted in the bottom frame of the DUT interface. When the top frame is clamped shut, the pogo pins make electrical contact with pads on the under-side of the DUT board. The pogo pins are mounted in blocks, in the bottom frame.

Connections From I/O Channels to Pogo Pins

Figure 2-3 and Figure 2-4 show the conventional connection of the pogo pins to the mainframe I/O channels. The DUT interface of the Miniframe uses the same layout convention, but has no space for the pogo pins of the extender mainframes.

Note 👉 You can, of course, connect your I/O channels to pogo pins in a *different* configuration to Figure 2-3 and Figure 2-4, but this could cause you considerable confusion when you try to diagnose a wiring fault!

CC = CARD-CAGE   BD = BOARD   CH = CHANNEL
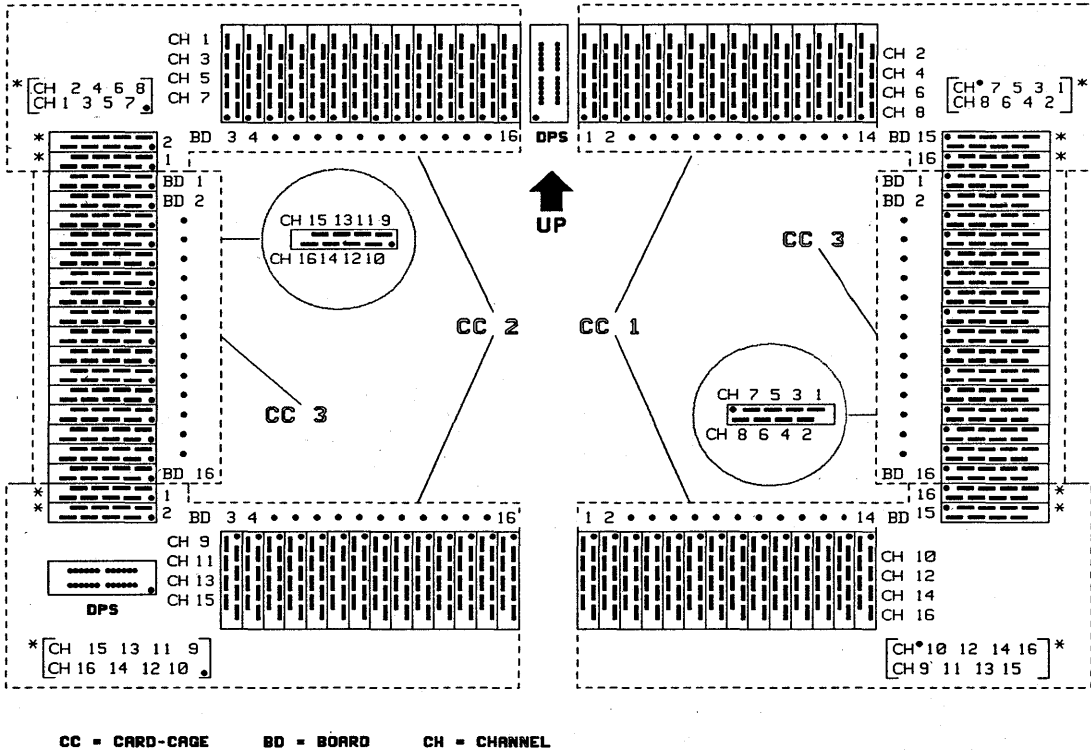
Figure 2-3. Pogo Pin Layout in the Bottom Frame of D50 Systems
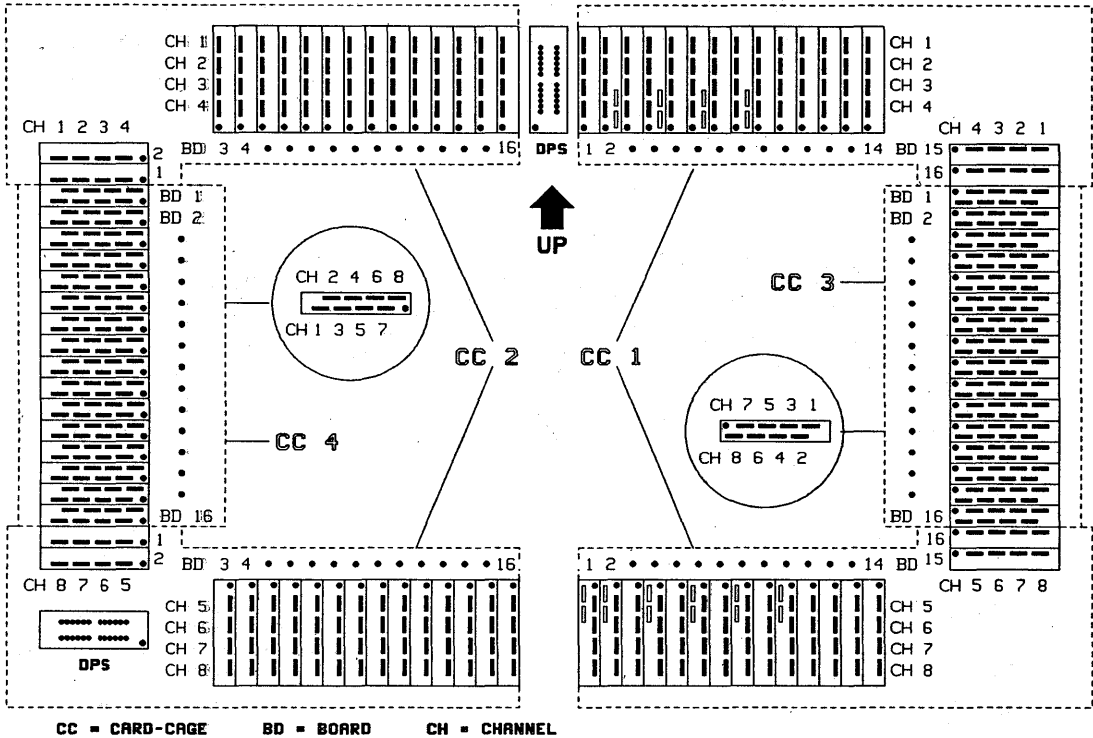
Figure 2-4. Pogo Pin Layout in the Bottom Frame of D100, D200 and D400 Systems

---

Connecting Device Power Supplies

Setting DPS Polarity

Figure 2-5 shows the polarity switching pads for the power supply DPS1, located in one corner of your DUT board. You set the polarity by making two soldered links. The illustration (Figure 2-5) shows how to connect DPS1 to have a **POSITIVE** supply and **NEGATIVE** ground connections.
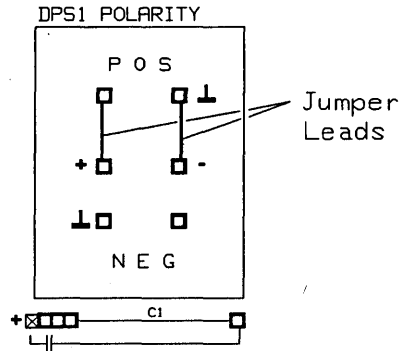
Figure 2-5. Selecting DPS1 Polarity

To select negative polarity, insert the links from the +
and - terminals to the lower pair of holes.

Pads for setting the polarity of DPS2 to DPS8 are also
labeled on the DUT board, and you can set them in
exactly the same way.

Connecting the Power Planes

Two power planes (P1 and P2) are contained within the
layers of the DUT board. You can connect any two of
the DPS units DPS1, DPS2, DPS3, or DPS4, to these
planes.

If you need to use more than two DPS units, you can
directly connect these to your DUT socket using a wire
link, soldered from the DPS output pad to your pin grid
or DIL socket.

With pre-wired PGA boards, additional pads connected
to P1 and P2 are located *within* the pin grid. This
allows you to connect any pin in your PGA test-socket
to either of the power planes independently.

Using the High Current Connections for DPS1 and DPS2

For some applications, the current that you need to supply to your DUT may exceed the maximum current that can carried by the pogo pins. For such applications, you can fit additional High Current connectors for supplying power to the DUT board from DPS1 and DPS2.

The connector fits in the top right hand corner of the DUT board, and the connections are identified by a silk-screen printed legend.

The DUT board accepts a *MATE-N-LOK* type connector, manufactured by **AMP**. You must wire the other end of the High Current connector to the DPS, in parallel with the connections to the pogo pins.

The maximum current that can be carried by the pogo pins is 5.0 A. With high-current connectors fitted to your DUT board, you can use DPS1 and DPS2 to each supply up to 10.0 A to your DUT.

Connecting the DC Rings

The DUT Boards have four DC Rings (DC1, DC2, DC3 and DC4) which run along the edges of the I/O channel pads. You can use these rings to connect pull-up or pull-down resistiors to the tester I/O channels, with the minimum of wiring. The four dc rings may be connected to DPS1, DPS2, DPS3, and DPS4.

For example, Figure 2-6 shows DC Ring 3 connected to DPS3.

Figure 2-6. DC Ring Connections
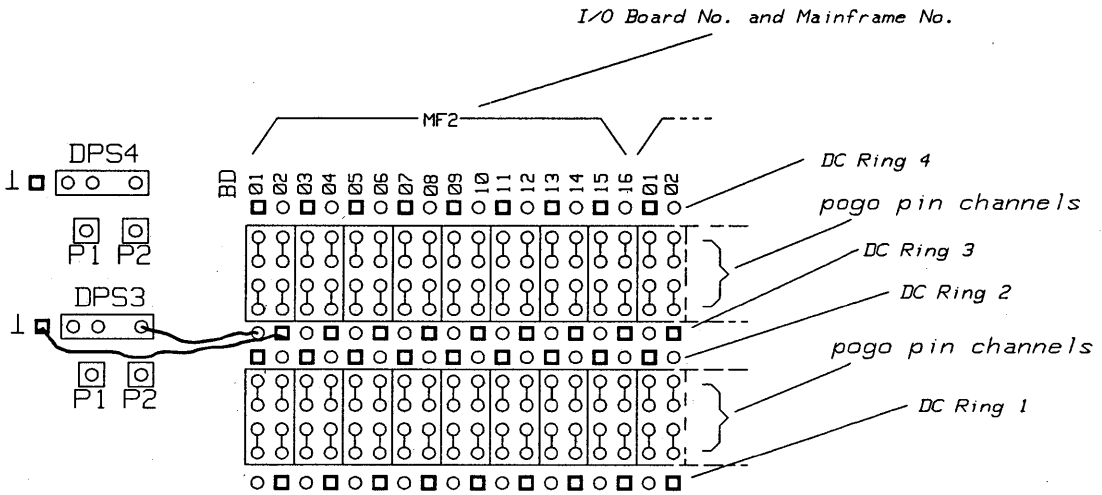
□ indicates a pad connected to Ground

O indicates a pad connected to power

High Frequency Decoupling

To remove high frequency noise in the power supply to the DUT, you must mount power supply decoupling capacitors between each power pin and ground.

The DUT board has mounting pads for connecting electrolytic capacitors to each DPS. These mounting pads are located near the polarity setting pads of each DPS, and are labelled:

C11	for DPS1
C21	for DPS2
C31	for DPS3
C41	for DPS4
C51	for DPS5
C61	for DPS6

C71   for DPS7
C81   for DPS8

The electrolytic capacitor should be in the range: $10\mu F$ to $100\mu F$. The actual value of the capacitance that you need depends on the frequency of the noise, and value of the current.

On some DUT Boards, pads are available for mounting n ceramic capacitors. The pads are numbered as CXn, where X = DPS number (1 to 8) and n = capacitor number.

Suggested values for the capacitor are:

- $0.1\mu F$
- $0.047\mu F$
- $0.01\mu F$

You must also mount additional ceramic capacitors directly below the DUT. The pre-wired DUT boards have signal ground pads, within the pins of your test socket, for this purpose. Your capacitors should be connected between the DUT power pins and these ground pads, and you should keep the leads as short as possible.

Connecting HSWG Channels (D400 Systems)

Because of the very fast edge-transition speeds of the High Speed Width Generator (HSWG) channels, you must take special precautions when you wire these channels on your DUT board.

Coaxial Wiring

Use miniature coaxial cable for all the connections on the DUT board which carry HSWG signals; avoid using the printed circuit wiring within the DUT board. Try to keep the cable lengths as short as possible.

HSWG Pogo Pads

Special connection pad areas are positioned directly beside the the pogo pin contact pads on the under-side (pogo pin side) of the DUT board. These pads allow you to wire your HSWG channels to your DUT, directly from the point where the pogo pin makes contact with the DUT board.

Use coaxial cable to make the connections from these pads directly to the socket in which you will test your DUT. The pogo pads carrying HSWG channels *are not* internally tracked to the pad fields beside the DUT socket.

Figure 2-7 shows the layout of the pads on the under-side of the DUT board, for connecting three of the HSWG channels. Silk-screen printed lettering on the DUT board itself identifies which pads are connected to which HSWG channels.
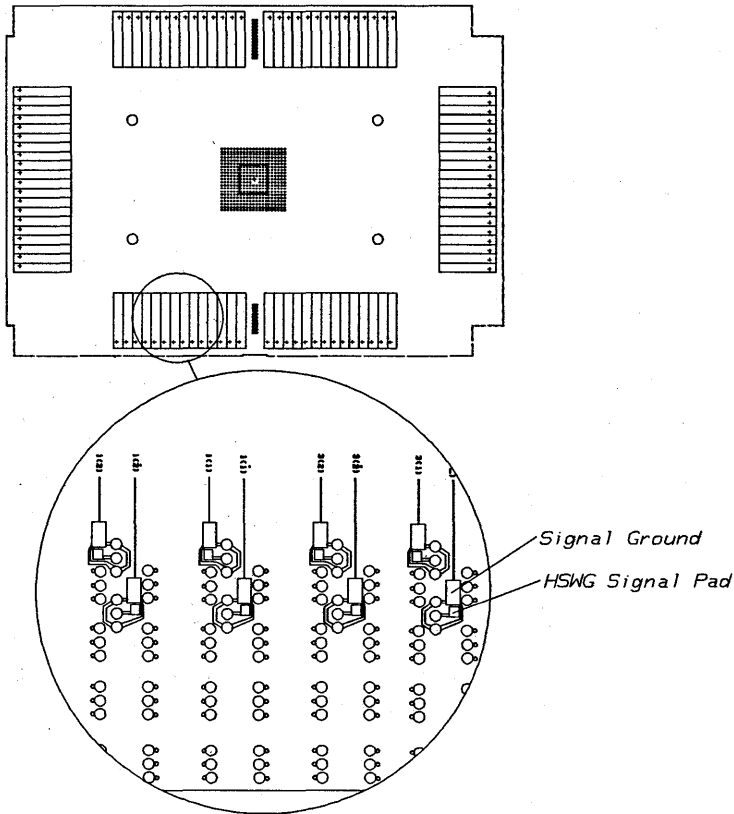
Figure 2-7. Under-side of DUT board, showing HSWG Wiring Pads

Stub Lines   With your HSWG signals, ensure that there are no
internally tracked lines left open ended, on your DUT
board. If your DUT board is of the pre-wired type, you
may have to electrically isolate the pins of your device
from the DUT board, by either:

■ drilling out the contact pad in the Pin Grid,

  or

■ disconnecting the pin of the DUT socket from the
  DUT board.

Fitting Additional Components

For some types of testing you may need to fit additional components onto the DUT board. These components could be:

- Series Resistors (either to form a resistive divider arrangement, or to provide a 50 Ω termination);
- Test Sockets (Scope Probe etc);
- Additional components, for example:
  - □ SIL Resistor Packages
  - □ Capacitors
  - □ Other discrete or SIL-packaged components

Note

For indicating the functions of DUT board pads, the illustrations in this section use the following convention:

- A diamond symbol designates a pad connected to the DC ring
- A square designates a GND connection
- A circle designates a signal pad

These markings are not always printed on the DUT boards

Using Series Resistors (R$_s$)

Under some circumstances, you may need to add a resistance in series with the signal going to or from your DUT, for example, to provide

- a termination resistor for impedance-matching the transmission line created by the pin driver electronics of the tester
- or a resistive divider arrangement for testing low-current DUT outputs at high frequencies (especially CMOS devices).

A full discussion of these techniques, and the considerations that you must make when laying out your DUT board, are given in the reference guide *High Speed Testing*.

Fitting Serial Resistors to the DUT Board

The DUT boards have a pad field printed into the tracking between the pogo connection pads and the test socket pins. If you are using a standard DUT board, you will have to make the wire connections between the Rs pads and the test socket pins yourself.

If you are using a pre-wired board, you do not have to make any connections between the R_s_ pads and the test socket pins: the connections are made through the printed wiring of the DUT board.

To fit a series resistor to your DUT board complete the following procedure:

1. Locate the series pads of the channel in which you intend to insert a resistor. The two series pads are linked by a strip of printed circuit track (Figure 2-8).

128/64 PIN BOARD   256 PIN BOARD   > 256 PIN BO

◇ DC   □ GND   CHANNEL
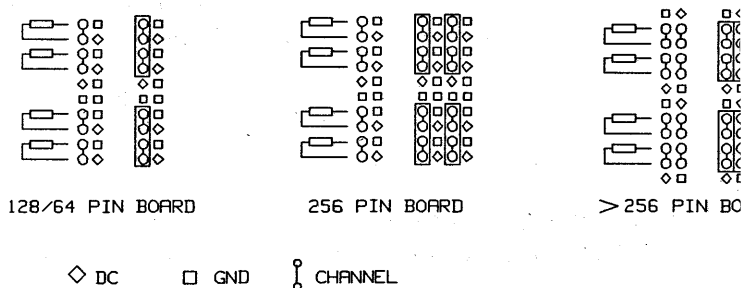
Figure 2-8. Mounting a Series Resistor

2. Using a scalpel, cut the printed circuit track linking the two pads.
3. Solder the resistor into position.
4. Using a wire link, connect the open end of the resistor to the appropriate DUT pin connector.
5. Clean all excess solder/flux from the area of the soldered joints.
6. Do not forget to make the relevant entries in the Pin Attributes Setup window if you have inserted any

serial resistors on the DUT Board. Full details about using the Pin Attributes Setup window, are given in the manual *Advanced Testing with the HP 82000.*

Fitting Test Sockets

If you are monitoring a channel using an external instrument (for example an oscilloscope) a test socket should be fitted to the appropriate channel pad. This will allow you to easily connect your oscilloscope probe.

Note

When monitoring a system channel using external equipment such as an oscilloscope, consider the effects of capacitive and resistive loading.

To install a test socket on a DUT board complete the next procedure.

Note

For 64/128 pin boards, every channel pad (in a pad field) will have **1 Ground Contact Pad.**

For 256 pin boards, every channel pad (in a pad field) will have **1 Ground Contact Pad.**

For DUT boards with more than 256 pins, there is only **one ground contact pad for every two channels.** This means you can only install a test socket on every other channel.

1. Locate the R_s pad (channel) in which you intend to mount a test socket.
2. Refer to Figure 2-9, remove one ground connector from the test socket.

every Channel
1 GND Contact
128/64 PIN BOARD

every Channel
1 GND Contact
256 PIN BOARD

every second
Channel (1 GND)
>256 PIN BOARD

◇ DC   □ GND   CHANNEL

Test Socket
Cut off one
GND Connectio
Channel Connection
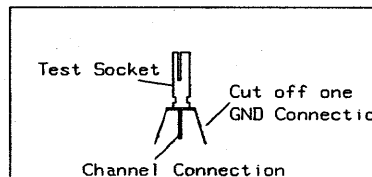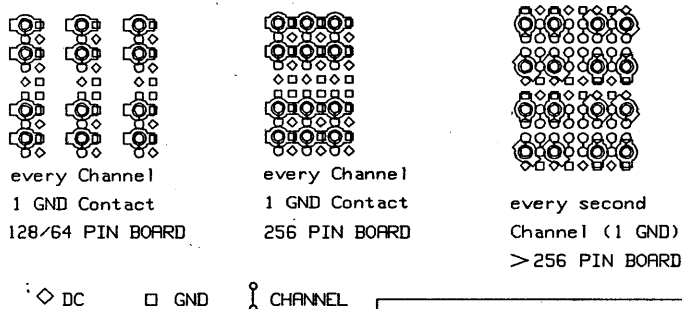
Figure 2-9. Fitting a Test Socket

3. Position the socket on the DUT board, the center channel connector through channel pad, the ground connector to ground.
4. Solder the test socket into position on the rear of the board.
5. Clean all excess solder/flux from the area of the soldered joints.

Fitting SIL Packages

When you are fitting SIL packages, take care to connect the **GND** and **DC** pins to the correct pads. Refer to Figure 2-10.

128/64 PIN BOARD   256 PIN BOARD   > 256 PIN BOARD

◇ DC   □ GND   ⟂ CHANNEL
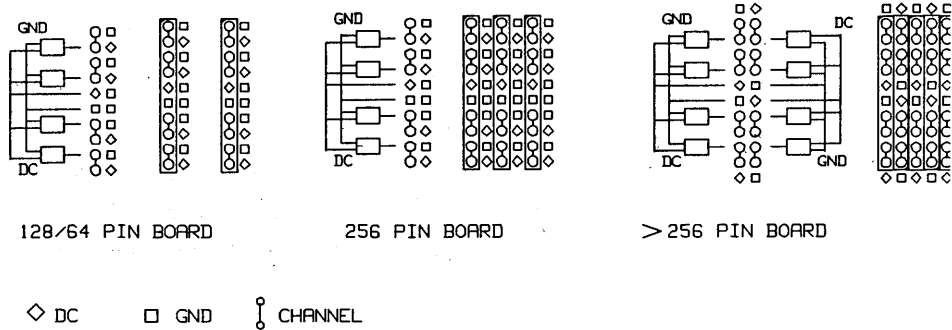
Figure 2-10. Fitting a SIL Package

Fitting Discrete Components

Fit discrete components to your DUT board as shown in Figure 2-11.

128/64 PIN BOARD   256 PIN BOARD   > 256 PIN BOARD

◇ DC   □ GND   ⟂ CHANNEL
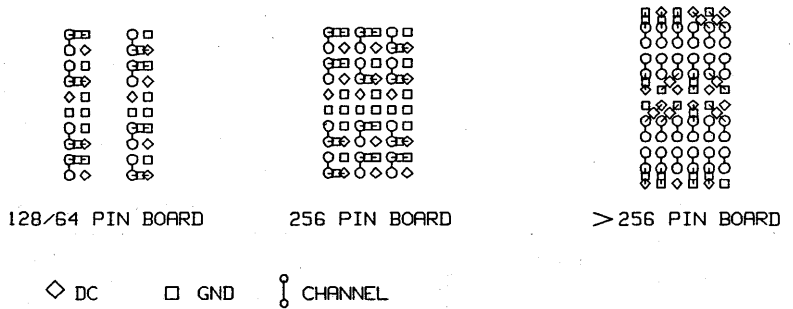
Figure 2-11. Fitting Discrete Components

Utility Lines   To enable connection to other equipment, or for remotely
controlling other components (e.g. relays) fitted to your
DUT board, you can fit an additional pogo pin block
and cable set for operating the utility lines.

The utility lines are connected to the sequencer board
via a pogo pin block installed in the pogo pin frame and
can be controlled by the system software. Figure 2-12
shows the utility line pads on the DUT board.

```
                          O O
                           O O     +5 V
       -5.2 V            O O
                           O O    GND
                  I3     O O
                  I1     O  O     I2
                  O6     O  O     I0
                  O4     O  O     O5
                  O2     O  O     O3
                  O0     O  O     O1
```
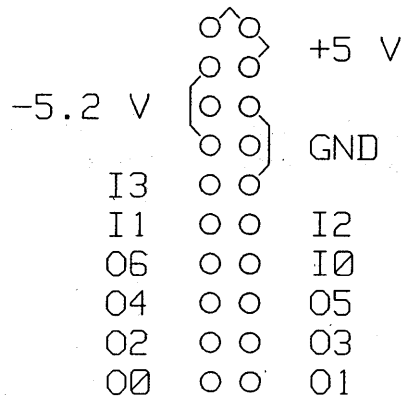
Figure 2-12. Utility Line Pads.

Note   Ensure that the soldered wire-ends which protrude
through the utility line pads, are cut down as close
to the surface of the DUT board as possible, and are
covered over with insulating tape.

Because of the tightness of the seal between the top and
bottom halves of the DUT interface when it is clamped
shut, there is a possibility of a short-circuit in the utility
lines if they are not properly insulated.

Customized DUT Board Services

A customizing service is available for DUT board design and layout. Contact your local HP representative for details of customized DUT boards.

3
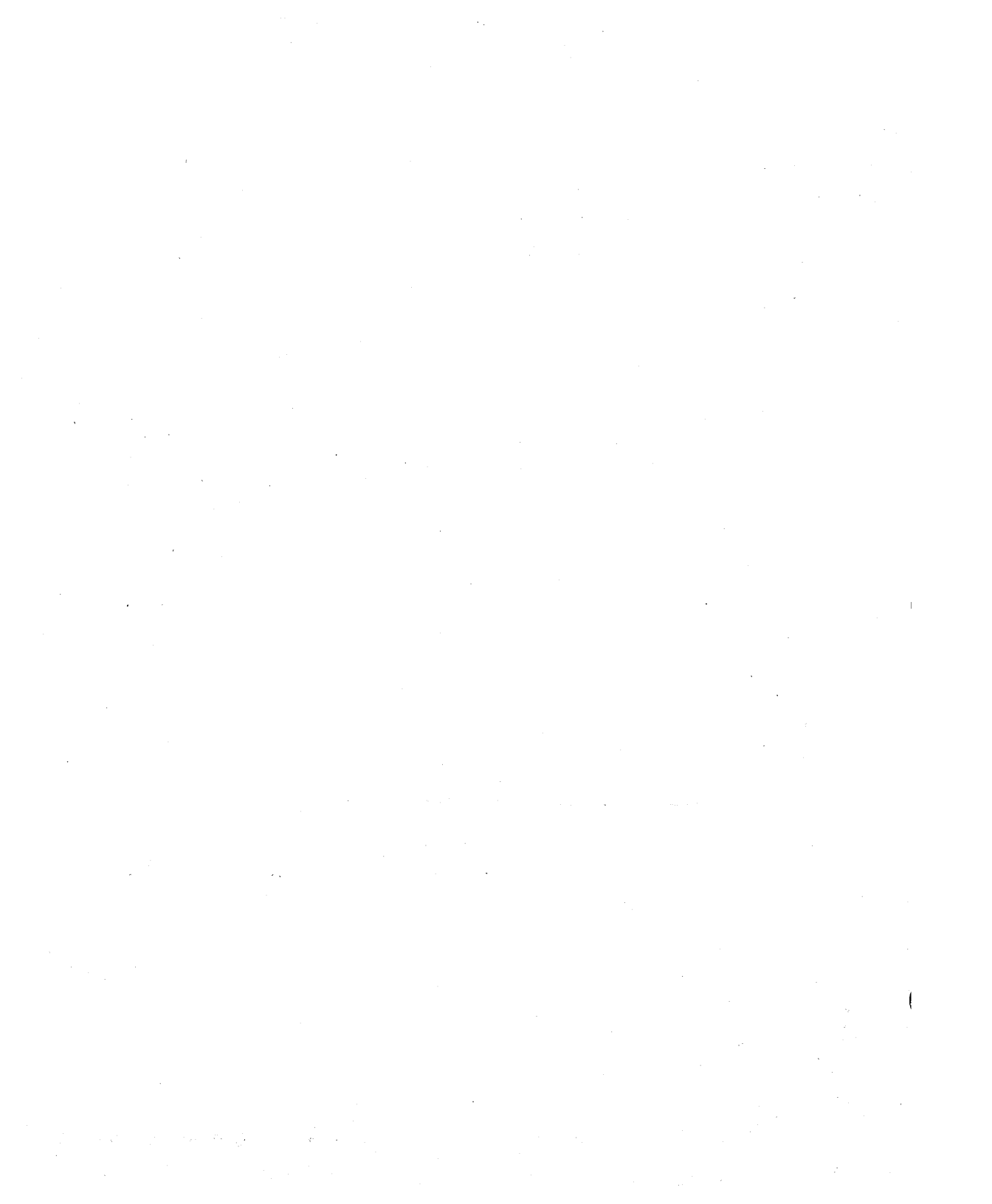
Configuring Pins for Testing

Using the Pin Configuration Window

The Pin Configuration window is where you allocate the resources of the HP 82000 to the pins of the device and define names for the pins. The configuration procedure, for each pin of the DUT, includes the following steps:

1. Set the Pin Configuration window to EDITOR mode

2. Specify the name of the DUT pin

3. Specify a pin number (optional)

4. Define the type of pin (e.g. input, output, power supply, etc ...)

5. Set the operating mode of the tester channel (e.g. 100, 200, 200MUX, 400, 400MUX, etc ...)

6. Map the DUT pin to a tester channel

7. Download this data to the tester hardware by clicking the (download) button.

Figure 3-1 shows the Pin Configuration window. The configuration displayed here is supplied to you as part of the system software. It is the configuration for the demonstration device, an MC10136 hexadecimal counter.

The data for configuring the pins of your device, is usually taken directly from your CAE system by the EDA Interface software. This data requires only minor adjustment.

If you are configuring a device *without* using simulation data from the EDA interface, you will need to enter all the configuration data into this window.

All other setups are subordinate to the pin configuration setup.

☒			▐▐ Pin Configuration ▐▐			

File Edit Select Mode

EDITOR Mode   Conversion OFF   (Download)

PIN			ATTRIBUTES		TESTER
Number	Name	Type	Mode	Scan	CHANNEL
13	Clock	i	100	par	10104
12	D0	i	100	par	10201
11	D1	i	100	par	10202
6	D2	i	100	par	10206
5	D3	i	100	par	10205
10	NCin	i	100	par	10203
4	NCout	ot	100	par	10108
14	Q0	ot	100	par	10101
15	Q1	ot	100	par	10102
2	Q2	ot	100	par	10106
3	Q3	ot	100	par	10107
9	S1	i	100	par	10204
7	S2	i	100	par	10207
1	Vcc1	dc	100	par	10105

Figure 3-1. Pin Configuration Window

Note

If you are manually typing data into this, or any other, setup window, it is a good idea to save your data at several stages during your session.

This is easily achieved by clicking the **save** or **save as** option in the `File` pulldown menu.

This practice will make it much easier for you to correct mistakes and recover accidentally erased data, and will save you from a lot of frustration in the long term!

EDITOR Mode and MONITOR Mode

As a precaution against unintended configuration changes, the Pin Configuration window has an EDITOR mode, which you must explicitly select before making any setup changes.

You can toggle between EDITOR and MONITOR Modes in the Mode pull-down menu. The current mode is displayed

in the top left-hand corner of the Pin Configuration
window.

Pin Names

The name that you enter in this field is the name that
the software uses to look up all attributes related to the
pin. You can use up to 16 alphanumeric characters in
the pin name.

In other windows, (such as the Level, Vector and
Timing Setup windows and the result windows) all
the information for the device pins is referenced to the
pin name. This allows you to refer to the pins using
meaningful names (such as "clock", or "Vee") instead of
having to remember all the tester channels that you have
used.

Pin Number

In addition to the pin name, you can also define the
number of the pin, with respect to the packaging of
your DUT. The number in this field is for your ease of
reference; it is not used by the tester.

You can use the pin number field to give you a quick
indication of the functions of the DUT pins, and their
positions, without constantly referring to data sheets.
This is particularly useful when you attach a probe or
oscilloscope lead to your device.

Pin Types

The pin type determines what hardware resources will be connected to a device pin.

Each pin of the DUT can be of one of the following types:

- input
- input (high-speed)
- output
- output (terminated)
- bidirectional
- bidirectional (terminated high)
- bidirectional (terminated low)
- dc measurement (PMU)
- disconnected
- dc power (DPS)

Note 👆 The pin types refer to the DUT, not to the HP 82000—so an *input* pin on the DUT is driven by the tester channel; the tester receives data from an *output* pin on the DUT.

Input Pins (i)

These pins are connected to the drive circuitry on the tester I/O channel. A drive channel basically consists of a drive pattern memory, timing and formatting circuitry, and pin driver circuitry. A simplified diagram of a pin's driver circuitry is shown in Figure 3-2.
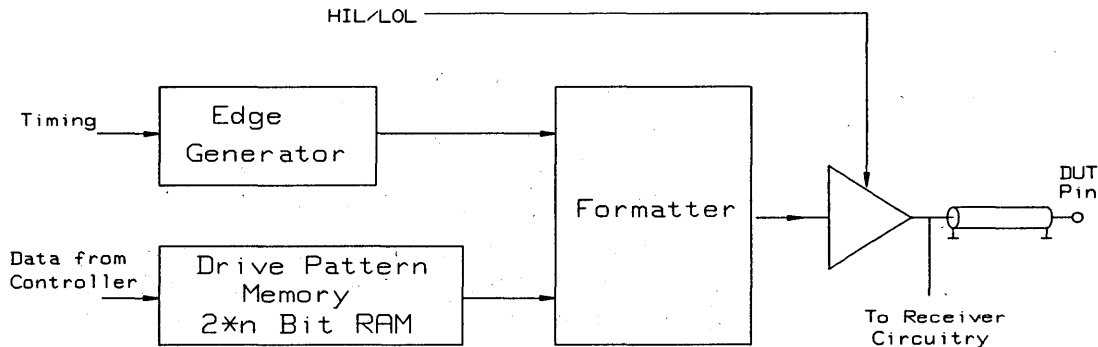
Figure 3-2. Driver Circuitry - Simplified Block Diagram

The drive pattern memory stores the pulse sequence to be used to drive the pin. The memory is organized as a $2 \times n$ Bit RAM, where n is:

for the D50   32K or 128K

for the D100   64K or 256K
and D200

for the D400   128K or 512K

Before the test starts, drive data (defined in the Vector Setup window) is downloaded from the workstation to the pattern memory. The data is fed out to the formatting circuitry at a rate determined by the timing period, set in the Timing Setup window.

The formatter also takes the timing parameters from the Timing Setup Window (leading and trailing edges), and combines them with the vector data to generate a train of pulses for the pin.

Level information, which is set in the Level Setup Window, is processed by the pin driver to set the pin's logic levels. Depending on how you terminate the pin on the DUT interface board, the output of the driver

is adjusted so that the level at the pin agrees with the value you have set in the Level Setup window.

Input High-Speed (ihs)

These pins are also connected to the driver circuitry of the I/O channel, and by using a High-Speed Width Generator (HSWG) unit, additional signal formats are made available for pins operating in 400 mode.

The pin driver is configured in exactly the same way as for the i pin, but instead of being connected directly to the DUT interface. the output is fed via the HSWG. The HSWG provides two, complementary, outputs.

Input High-Speed pins are only available on the D400 models. Details of the data formats made available by this pin-type, are discussed later in this chapter.

Output Pins (o)

In contrast to input pins, DUT output pins are connected to the receive circuitry of an I/O channel. The receive circuitry consists of a dual level comparator, a sampler, and a receive pattern memory (refer to Figure 3-3).
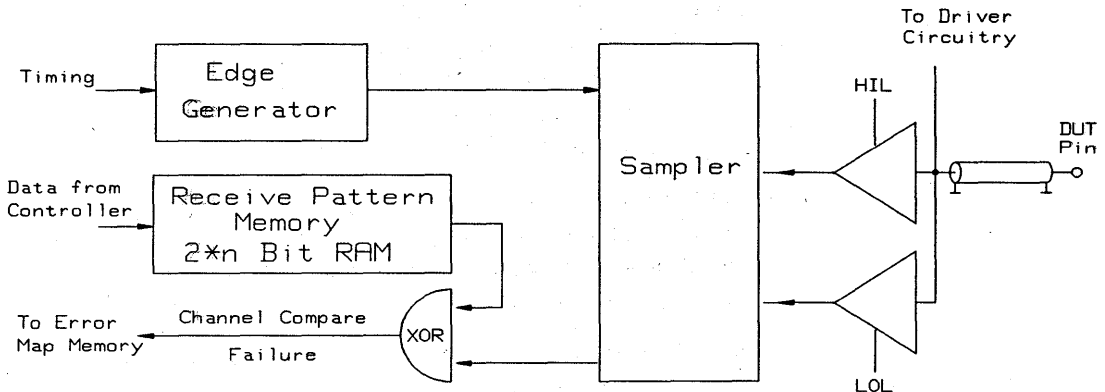
Figure 3-3. Simplified Block Diagram of Receiver Circuit

Like the drive pattern memory, the receive pattern memory is organized as a 2×n Bit RAM, where n is:

for the D50   32K or 128K

for the D100   64K or 256K
and D200

for the D400   128K or 512K

As data is being received from the output pin, it is continuously compared with the threshold levels that you have set in the Level Setup window. The comparison results are fed to a sampler. The timing parameters you set in the Timing Setup window, determine at what point in the cycle period the sampler measures the two comparator outputs.

The results of the comparisons are handled differently, depending on the *comparison mode* of the tester: either Data Acquisition (DA) mode or Real Time Compare (RTC) mode.

The comparison modes are described in detail in Chapter 7.

Terminated Output Pins (ot)

With these pins, the receiver resource is connected in the same way as for o pins, but instead of totally disconnecting the driver resource, the driver provides a constant, active-termination voltage. No drive data is sent from the vector pattern memory. Figure 3-4 shows an equivalent circuit for the ot pin type.
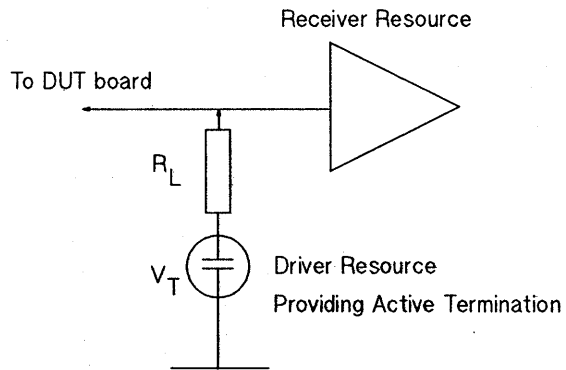
Where:

V_T = termination voltage set in Level Setup window

R_L = driver impedance (100 ohm – D50; 50 ohm – D100, D200, D400)

Figure 3-4. Equivalent Circuit for an ot Pin

When you select this pin type, the pin name appears twice in the Level Setup window:

- once as an o pin, allowing you to set the comparison levels for the pin;
- and once as type t, allowing you to set the termination voltage (V_T in Figure 3-4).

A full discussion of the application of active termination and resistive divider arrangements is given in the reference guide *High Speed Testing*.

Bidirectional Pins (io,ioh and iol)   When you select these pin types, the DUT pin has access to both the driver and receiver resources of the tester I/O channel. The Timing Setup, Level Setup, and Vector Setup windows list the pin name twice: once as an input to your DUT (the driver resource) and again as an output from the DUT (the receiver resource).

Because the driver resource is still connected to the pin
while the receiver is making its measurement, the output
impedance of the driver circuitry (100 Ω in the D50
and 50 Ω in the D100, D200 and D400) may affect the
measurement that you are making. You can prevent this
by ensuring that during the output cycle of the DUT
pin, your test vector sets the driver to:

- tristate (i.e. high-impedance);
- or (if tristate is not available) terminated to logic 0 or
  logic 1 by selecting ioh or iol.

The pin types ioh and iol allow you to use the driver
resource to provide an active termination voltage while
the DUT pin is in its output cycle. So while the DUT
pin is in its *input* cycle, you can use the driver to send
drive data (specified in the vector table); and while the
pin is in its *output* cycle, the tester is configured as an
ot type pin, with the termination voltage taken from:

- the driver logic high level, for ioh type pins;
- or the driver logic low level, for iol type pins.

Bidirectional pins can be used with D50, D100 and D200
Systems, and with 200MHz I/O boards installed in D400
systems.

A full discussion of the application of active termination
and resistive divider arrangements is given in the
reference guide *High Speed Testing*.

PMU DC Measurement Pins (dc)

These pins are used for DC measurement. The pin is not
connected to the drive or receive circuitry, instead, it is
connected to one of the Parametric Measurement Units
(PMUs).

**Disconnected Pins
(nc)**

These DUT pins are temporarily disconnected from the tester. However, the driver and receiver are allocated, and can be setup in the other windows.

nc pins can be used with D50, D100 and D200 Systems, and with 200MHz I/O boards installed in D400 systems.

**Device Power Supply
Pins (DPS+ and DPS-)**

These pins connect the Device Power Supply unit to the power pins of the DUT. There are no driver or receiver resources connected to these pins.

Pin Operating
Modes

Every channel of each of I/O board can operate in one of several modes, which you enter in the **Mode** column of the Pin Configuration window. The mode you choose sets the configuration of the driver and receiver channels, and thus determines:

- the maximum frequency you can use for your tests;
- the signal formats with which you can drive your pins;
- the number of comparison thresholds that you can set, for measuring the DUT outputs;
- the type of comparison that is made;
- the number of DUT pins to which you can assign resources.

Memory Organization

It is important to realize how the tester firmware handles the vector data to understand the limitations that exist in some test modes. The organization of the drive and receive pattern memory is different for each mode, and this affects the portability of vector data across these modes. It also means that in some modes, certain signal formats are not available.

In the MUX Modes, two adjacent channels are combined together to form a single channel with double memory depth. Successive data bits are alternately taken from

the driver pattern memories of the two channels. This means that the data from one channel is interlaced with data from the one adjacent to produce the pin data, which doubles the effective data rate.

Input Pins   The following formats are available for driving the DUT input pins.

- Delayed non-return to zero (DNRZ)
- Delayed non-return to zero, with tristating (DNRZ + tristate)
- Return to zero (RZ)
- Return to one (R1)
- Return to complement (RC)
- Return to Inhibit (RI)

DNRZ + Tristate

On each alternate bit in the drive pattern memory, an edge is generated. The remaining bits are used to control the tristate capabilities of the pin driver. If the tristate bit is set, the pin driver will be driven to high impedance instead of the respective high or low level.

Data ───────────►   ─── High Z ───

Data Bits		0	0	0	0	0	1	0	1
64K ◄───────────		7	6	5	4	3	2	1	0
Tristate Bits		1	1	1	1	1	0	0	0
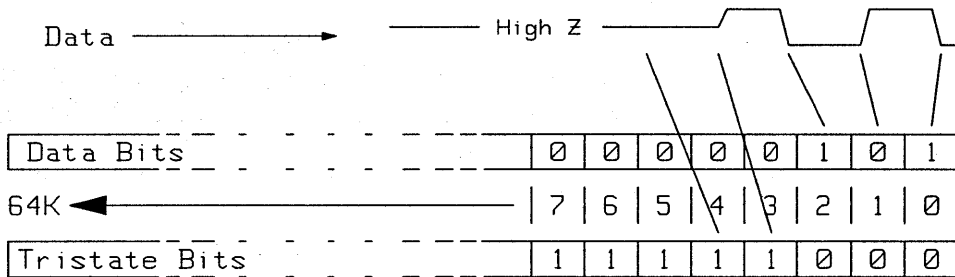
Driver Pattern Memory 2*64KBit

Figure 3-5. Memory Mapping - DNRZ + Tristate

DNRZ (no Tristate)

For this signal format, both of the drive pattern memory arrays store drive data for *each* cycle (instead of one array storing drive data while the second block stores tristate data). This gives an increased memory depth and gives an extra data bit for the second half of each clock cycle, thus doubling the data rate (although losing the facility of tristate).
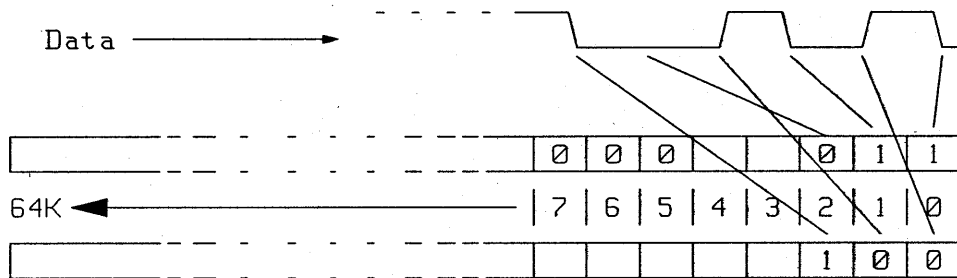
Driver Pattern Memory 2*64KBit

Figure 3-6. Memory Mapping - DNRZ (No Tristate)

RZ, R1, R0 and RC

With these formats, each alternate bit of the pattern memory is used to generate a leading edge, while the remaining bit generates a trailing edge. The leading and trailing edges derived from each bit are independently programmable (refer to the chapter on Timing Setup).

Data ⟶

64K

Driver Pattern Memory 2*64KBit
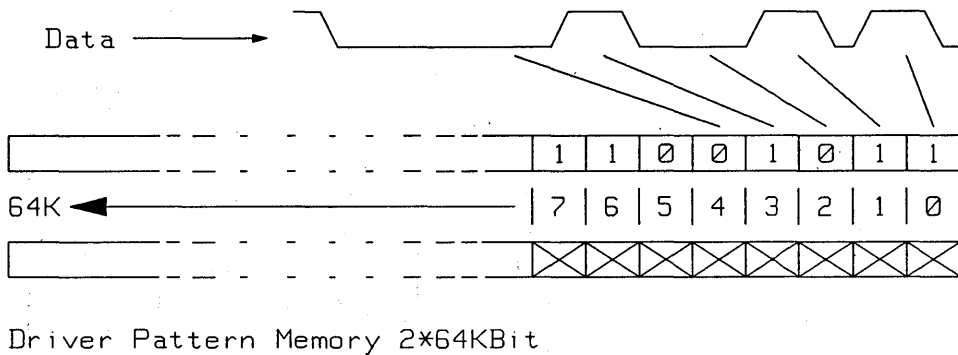
Figure 3-7. Memory Mapping - RZ, R1, R0 and RC

Output Pins   The receiver channels of the HP 82000 allow either window or edge strobing facilities.

Edge Compare Mode

The output from the DUT is measured at a set point in the cycle. The point can be delayed from the start of the cycle (for a more detailed explanation, refer to the chapter on Timing Setup). The result of an edge comparison can have one of three values; high, low, or intermediate.

Window Compare Mode

The DUT output is measured within a time window whose start and end points, relative to the start of a cycle, are set in the Timing Setup Window. The result of a window comparison has one of four possible values; high, low, intermediate, or unstable (glitch detected).

Availability of Signal Formats

The following section details the driver and receiver modes available with the different HP 82000 models.

D50 Systems

Table 3-1.
Modes and Permissible Formats—D50

	25	50	50MUX
DNRZ	yes	yes	yes
Tristate	yes	no	no
RZ	yes	no	yes
R1	yes	no	yes
RC	yes	no	no
RI	yes	no	no
Memory depth	standard	double	double
Channels needed per pin	1	1	2
permitted pin types	all except ihs	all except ihs	all except ihs

25 Mode

These channels operate at a maximum data rate of 25 MHz. In this mode, each alternate bit of the drive pattern memory is used to generate a signal edge, while the remaining bits are used to switch the driver to tristate.

Driver Formats

The signal formats and tristate capabilities available are:

- Delayed Non-Return to Zero, with Tristate facility (DNRZ + tristate)
- Return to Zero (RZ)
- Return to One (R1)
- Return to Complement (RC)
- Return to Inhibit (RI)

Receivers

For each receive channel in the Receive Pattern Memory, two bits per cycle are allocated, which results in up to four values.

This allows both dual-level edge compare, and window compare modes. Bit masking is also possible with this mode.

50 Mode

These channels operate at a maximum data rate of 50 MHz. The increase in speed is achieved by using the tristate part of the drive pattern memory to generate an extra edge in each period. Consequently, there is no tristate capability for driver channels in this mode.

Double the number of vectors are available in the Vector Table.

Driver Formats

Only DNRZ format is available.

Receivers

The receiver channels in this mode have only one bit per cycle. Thus, only a single-threshold edge comparison can be made per cycle.

Bit-mask does not operate in 50 mode.

50MUX Mode

This mode achieves the 50 MHz data rate by multiplexing together the driver and receiver resources from two adjacent channels. This doubles the memory depth available for the pin and enables you to use all data formats except tristate.

Each pin in the MUX mode requires the resources of two drivers and/or receivers. If you are using a bidirectional pin in MUX mode, the second channel of the MUX pair cannot be used.

However, if you are using an input pin (i) in MUX
mode, the other channel of the pair can be configured
as a separate output pin (o), using any of the modes
available on the machine. Similarly, if you are using
an o pin in MUX, the driver resources are free for
configuration elsewhere as an i pin. More details about
allocating the tester channels are given later in this
chapter.

Driver Formats

In this operating mode, all data formats are available
except for those using tristate, and return-to-complement
(RC).

The formats available are:

- DNRZ
- RZ
- R1

Receiver

On the receiver side, full dual-level edge compare and
window compare, are available, as well as bitmask.

D100 Systems

Table 3-2.
Modes and Permissible Formats—D100

	100
DNRZ	yes
Tristate	yes
RZ	yes
R1	yes
RC	yes
Memory depth	standard
Channels needed per pin	1
permitted pin types	all except **ihs**

100 Mode These channels operate at a maximum data rate of 100 MHz. In this mode, each alternate bit of the drive pattern memory is used to generate a signal edge, while the remaining bits are used to switch the driver to tristate.

Driver Formats

The signal formats available are:

■ DNRZ + tristate
■ RZ
■ R1
■ RC

Receivers

For each receive channel in the Receive Pattern Memory, two bits per cycle are allocated, which results in up to four values.

This allows dual-level edge compare and window compare modes. Bit masking is available in this mode.

D200 Systems

Table 3-3.
Modes and Permissible Formats—D200

	100	200	200MUX
DNRZ	yes	yes	yes
Tristate	yes	no	no
RZ	yes	no	yes
R1	yes	no	yes
RC	yes	no	no
Memory depth	standard	double	double
Channels needed per pin	1	1	2
permitted pin types	all except ihs	all except ihs	all except ihs

100 Mode These channels operate at a maximum data rate of 100 MHz. In this mode, each alternate bit of the drive pattern memory is used to generate a signal edge, while the remaining bits are used to switch the driver to tristate.

Driver Formats

The signal formats available are:

- DNRZ + tristate
- RZ
- R1
- RC

Receivers

For each receive channel in the Receive Pattern Memory, two bits per cycle are allocated, which results in up to four values.

This allows dual-level edge compare and window compare modes. Bit masking is available in this mode.

200 Mode

These channels operate at a maximum data rate of 200 MHz. The increase in speed is achieved by using the tristate part of the drive pattern memory to generate an extra edge in each period. Consequently, there is no tristate capability for driver channels in this mode.

Double the number of vectors are available in the Vector Table.

Driver Formats

Only DNRZ format is available.

Receivers

The receiver channels in this mode have only one bit per cycle. Thus, only a single-threshold edge comparison can be made per cycle.

Bit-mask does not operate in 200 mode.

200MUX Mode

This mode achieves the 200 MHz data rate by multiplexing together the driver and receiver resources from two adjacent channels. This doubles the memory depth available for the pin and enables you to use all data formats except tristate.

Each pin in the MUX mode requires the resources of two drivers and/or receivers. If you are using a bidirectional pin in MUX mode, the second channel of the MUX pair cannot be used.

However, if you are using an input pin (i) in MUX mode, the other channel of the pair can be configured as a separate output pin (o), using any of the modes available on the machine. Similarly, if you are using an o pin in MUX, the driver resources are free for configuration elsewhere as an i pin. More details about allocating the tester channels are given later in this chapter.

Driver Formats

In this operating mode, all data formats are available except for tristate, and return-to-complement (RC).

The formats available are:

- DNRZ
- RZ
- R1

Receivers

On the receiver side, full dual-level edge compare and window compare, are available, as well as bitmask.

D400 Systems

Table 3-4.
Modes and Permissible Formats—D400

	200COM	400	400MUX
DNRZ	yes	yes	—
RZ	yes	yes*	—
R1	yes	no	—
complement	yes*	yes*	—
Memory depth	standard	double	double
Channels needed per pin	1	1	2
permitted pin types	i,ihs,o,ot	i,ihs,o,ot	o,ot

*when used with ihs type pins.

200COM Mode

This is the 200 MHz, *compatibility* mode for the D400.
When you choose this mode, the D400 channels offer the maximum number of formats

This mode cannot be used with bidirectional pins.

Driver Formats

- DNRZ
- RZ
- R1
- Complementary output (when using ihs pin-type)

Receivers

The receiver channels use dual-level edge comparison.

400 Mode

This mode can operate in a similar way to the 200 mode in the D200 systems, to generate a DNRZ signal at a frequency of up to 400 MHz.

If you select the ihs pin type, you can also generate RZ format signals, and a complementary output.

400 mode cannot be used with bidirectional pins.

Driver Formats

- DNRZ
- RZ (when using ihs pins-type)
- Complementary output (when using ihs pin-type)

Receivers

Only single-threshold edge compare mode is available. Bit-mask is not possible in this mode.

400MUX Mode

This mode only operates with output (o) pins.

The 400 MHz data rate is achieved by multiplexing together the receiver resources from two adjacent channels. This enables a dual-level edge-compare and bit-masking capability.

Scan Path Testing Mode

Some digital logic devices are designed with the capability to test themselves, or to output status information about the internal circuitry. Normally, one or two pins on a device are dedicated to this function, and test data is input and output in serial form. These test vectors can be extremely long and often exceed the memory depth of the system.

To support this type of testing, called *Scan Path* testing, you can concatenate two or more channels into a single test channel.

Setting Scan Path Parameters

The default scan attribute for all pins is par (parallel). This means that each channel is connected to a DUT pin.

For Scan Path testing, you must serialize several channels by selecting ser from the fill menu. The other configuration parameters for the serial channels must be setup in the same manner as for parallel channels.

Setting up a Scan Path Test is rather more complex than just changing pin attributes. The whole subject of Scan Path testing is described in detail in the manual *Advanced Testing with the HP 82000*.

Word Mask

The word mask feature allows you to blank out an entire vector during a test. Errors occurring in this masked vector are ignored. Word mask is especially useful for masking pins configured in modes which cannot use bitmask.

To operate the word mask, you must set up a specially reserved I/O channel in the Pin Configuration window.

You can use this channel as any other channel, if word mask is not required. When word mask is required,

you have to dedicate the receiver side of this channel to word mask (although you can still use the driver side to provide another input pin).

Setting the Word Mask Channel

1. To help you remember that this channel is dedicated to word mask, enter the pin name as *word mask, w/mask,* or similar.
2. Set the pin type to o (output).
3. Set the operating mode to any non-MUX mode.
4. From the fill menu in the Scan column, select wmask.
5. Set the tester channel to
   - 10116 - if you are using a D50 system;
   - 10108 - if you are using a D100 or D200 system;
   - 10101 - if you are using a D400 system.

Using the Driver Side of the Word Mask Channel as Input Pin

If you are running short of channel resources, you can free the driver side of the word mask channel, for providing an additional input signal to your DUT. (This technique cannot be used with D400 systems).

1. Enter a pin name that reminds you that you have split the receiver and driver (for example, Input/Wmask).
2. Set the pin type to io
3. Set the operating mode to any non-MUX mode.
4. From the fill menu in the Scan column, select wmask.
5. Set the tester channel to
   - 10116 - if you are using a D50 system;
   - 10108 - if you are using a D100 or D200 system.

You can now see the driver listed separately in the Level Setup and Timing Setup windows, where you can define your driver signal.

In the vector window, you also see the Input/Wmask pin listed twice; once as drive data; and again as expect data. You must remember that

- the expect data bit is for selecting and deselecting word mask;

■ and the drive data is your DUT input-pin.

Mapping DUT Pins to I/O Channels

The final task to configure the pins of your DUT, is to allocate the tester's resources. This is the process of assigning the input and output channels of the I/O boards and the Device Power Supply, to the pins of your DUT.

Factors that you must consider, at this stage, are;

■ the number of channels installed in your system;
■ the availability of:
    Device Power Supply (DPS) units;
    High Speed Width Generators (HSWG);
■ the physical location of the I/O boards within the card cages;
■ the wiring of your device to the DUT interface board (described in Wiring the DUT Board);
■ the operating modes you have selected for your pins.

I/O Channel Identification

The I/O channels are mounted on removable I/O Boards, which fit into

■ card cages installed within Standardframe or Maxiframe cabinets;
■ Miniframe cabinets and Miniframe extender cabinets.

The software identifies each channel by using a unique identity number. This number identifies:

■ the number of the card cage;
■ the position of the I/O board;
■ and the channel number on that board.

This is the number that you use to identify your I/O channels when you fill in the TESTER CHANNEL column in the Pin Configuration window.

For example, the channel located within:

- card cage 1;
- the second I/O board from the left;
- channel number five on the I/O board;

has the channel number 10205.

There are eight I/O channels on each I/O Board of D100, D200 and D400 systems, and sixteen I/O channels on each I/O board of a D50 system.

For further details of the hardware channel configuration, see the manual *Installing the HP 82000.*

Powering the Device

If a pin on your DUT has been allocated as a device power supply pin (DPS+ or DPS-), you must enter the relevant DPS in the TESTER CHANNEL column.

Allocating I/O Channels to Pins Automatically

When you connect the tester hardware online, you can instruct the software to automatically determine which channels are available, and to map them to the pins you have entered in the Pin Configuration window.

In normal use, the system will allocate the lowest machine channel to the first pin, the next highest channel to the second pin, and so on.

If channel multiplexing (50MUX, 200MUX, or 400MUX mode) is specified for a pin, the system automatically allocates an even numbered I/O channel and the next odd-numbered channel to that pin.

Allocating Channels with the Calibration Probe

This is the most reliable method of assigning I/O channels to the the pins that you have configured. It allows you to identify the pin on the DUT board (that you have already wired) by touching it with the calibration probe. The software then assigns the tester I/O channel connected to that DUT pin, to the pin name in the Setup window.

For information about connecting the calibration probe, refer to the chapter about Calibration in the manual *Maintaining the HP 82000*.

Procedure for Allocating Channels with the Calibration Probe

Before you start to map your channels with the probe, ensure that the DUT board in the tester is the one that you wish to use. Then, go through the following steps:

Caution

there is no device inserted in the DUT Board socket.

1. click the mouse on (AUX) and (Change Dev) in the double row of keywords at the top of the screen

2. click the enter pushbutton in the System Administrator dialog box

3. overtype the path name entry field in the next dialog box with /hp82000/pws and hit the (Return) key

4. select dut_boards from the browser and click the enter device push button

5. carefully read the next dialog box. If you wish to save your existing setups, click the store 'LAST_SETTING' checkbox. Click the (continue) pushbutton. The system software closes all the windows.

6. click the mouse on (MAIN) and (Pin Config) in the double row of keywords at the top of the screen

7. enter the Monitor Mode by clicking the MONITOR option on the Mode pulldown menu.

8. discard any existing pin configuration by clicking the new option on the File pulldown menu

9. switch back into EDITOR mode.

10. enter all DUT pin numbers into the Number column

11. create an entry field over number 1 in the Number column

Ensure that the entries are in the same order that you will contact the pins with the probe. An ascending numerical order is the best.

Now click the Allocate TESTER CHANNELS option on the Select pulldown menu. A dialog box is displayed as shown in Figure 3-8. The dialog box appears over the Pin Configuration Window. (You can move it out of the way by clicking the mouse on its heading - allocate TESTER CHANNEL - and dragging it away).

```
┌──────────────────────────────────────────┐
│         - allocate TESTER CHANNEL -        │
│                                            │
│               !!! CAUTION !!!              │
│                                            │
│   Channel allocation causes voltages to    │
│   appear on the DUT board that could damage│
│   the DUT.                                 │
│                                            │
│   Remove the DUT and press continue to     │
│   carry on.                                │
│                                            │
│          ( Continue )   ( cancel )         │
└──────────────────────────────────────────┘
```
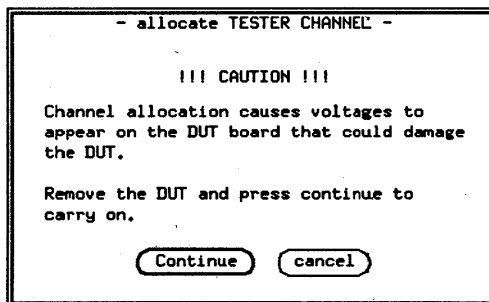
Figure 3-8. Mapping Pins to Channels - Dialog Box

Click the (Continue) pushbutton. The dialog box gives you more instructions. **Read them carefully.** You are now

about to go round the sockets in the order entered in the
Number column.

Make sure the probe's ground contact touches the gold
plated surface of the ground plane **before** you contact
the tip of the probe to each socket.

Take the probe and contact it onto the socket for the
first DUT pin. The system supplies the number of the
I/O channel in the **TESTER CHANNEL** column and moves
the entry field to the next row. The LED on the probe
lights. Go on to the socket for the second pin.

Repeat the procedure until all channels have been
mapped. When you get to the last channel the system
automatically exits the mapping routine.

If you make a mistake, for instance you skip a channel
or map a channel twice, you can break out of the
auto-mapping routine by clicking the mouse on the
(cancel) pushbutton. You then have to restart the routine
and finish the job. Allow the system to check for
duplicately mapped channels by clicking on the (Download)
pushbutton.

Now you have to store the created map file. Select the
save_as entry in the **File** pulldown menu and enter a
name for the device geometry you have just mapped.

Having finished mapping, you have to get back to your
old directory where your device files are stored. This will
probably be under **/users/logname**, where **logname** is
your log in directory. To get there, repeat the procedure
above. In the final step enter the path to your particular
device.

This method may seem long, but is considerably faster
than manually searching for the I/O channel numbers
on a DUT Board, equating them to DUT pin numbers
and entering the data, via the keyboard, into the Pin
Configuration Window or into a file.

Mapping DPS and Ground Pins

DPS and ground pins are to be found in different places even in devices of identical geometries. As far as the auto-mapping routine is concerned, a DPS or ground pin is just another pin to which a I/O channel is connected and as such must be included in the mapping.

When you have finished mapping pins, look at your device and decide where the DPS and ground pins will be. You then have to disconnect these pins from the I/O channels by cutting the track on a pre-wired DUT Board. Where to cut the track is explained in the Installation Manual. On a blank DUT Board you have to do all the wiring yourself and there is no need to wire up the DPS and ground pins to run the auto-mapping routine.

Other Methods of Mapping Pins to Channels

Manual Mapping

The DUT Board has markings on its surface, showing the standard positions of the pogo-pin connections to the I/O channels. (This information is correct providing you have not changed the positions of the pogo-pin blocks or mixed up the channel-to-DUT Board wiring in any way). From the existing wiring on pre-wired DUT Boards, or from your own wiring on blank boards you can determine the DUT pin number to I/O channel number mapping.

You enter these numbers into the Number and TESTER CHANNEL columns of the Pin Configuration Window. Alternatively, you can do this from the HP-UX prompt and create the required file directly using the vi editor and the necessary HP-IB programming commands.

This is best done by copying and suitably modifying an existing mapping file. HP-IB commands are listed in the *HP-IB Command Reference.*

Using a Mapping Program

Since the geometries of both DIL and PGA devices are laid out to standard conventions, you could write a short program in C or BASIC to generate the mapping file for you. Once written, it is the fastest method of all.

Using Standard Pin Mapping Files

For standard DUT pin packages, you can generate a mapping file that maps device pin numbers to I/O channel numbers, and DPS channels to the power pins of the device.

Once you generate a mapping file, you can use it for any device packaged with the same geometry and pin count. You can then generate another file to map the names, pin types and operating modes used by each individual device, to the standard mapping file for the package.

These files are displayed in two browsers, which you can find in the Select pulldown menu, under the map pin to channel option.

A mapping file has an ASCII format and resides in the /hp82000/pws/dut_boards/configuration directory. There are a number of ways to generate a mapping file.

Pin Configuration Files

Your device file directories will probably be in /users/logname/device_name. Thus the directory /users/demo/mc10136/configuration contains all pin configuration files for the device MC10136 under the login demo.

A pin configuration file can be generated by several means:

- by entering data into the Pin Configuration Window
- by creating the file at the HP-UX prompt with the vi editor
- by extracting the pin configuration from a simulation file (this is covered in the manual *Using the EDA Interface*)

A configuration file may not necessarily be complete. This means, not all columns in the Pin Configuration Window may contain data, or the entries may contain default data. This applies especially to configuration files extracted from simulation files, as the tester has no way of knowing how DUT pin names map to DUT pin numbers, and how pin numbers map to I/O channel numbers.

This information has to be supplied by you. Mapping of pin names to pin numbers comes from your layout design or from a device data sheet. Mapping of pin numbers to I/O channel numbers comes from the DUT Board layout and you obtain it with the calibration probe as described above.

Reusing the Same Channel Configuration

You could create an incomplete pin configuration file containing only DUT pin numbers and pin names, and use it on occasions when you want to test a whole family of similar devices with the same DUT Board and the same socket. Such a file is called a *template file*. In this case you don't want to allocate tester channels for each

device that you test. The system allows you to merge
pin configuration files together to save you this task.

To use the pin numbers and tester channels from another
configuration, click the **map pin to channel** option on
the **Select** pulldown menu. The system displays the
dialog box as shown in Figure 3-9.


```
┌─────────────────────────────────────────────────┐
│           - Map Pin Names to Tester Channels -   │
│                                                   │
│     Name -> Number    Number -> CHANNEL           │
│    ┌───────────────┐ ┌───────────────┐            │
│    │EDITOR_BUFFER  │ │DIL_16         │            │
│    │LAST_SETTING   │ │               │            │
│    │c_d200         │ │               │            │
│    │c_d200_hil     │ │               │            │
│    │c_d200_men     │ │               │            │
│    │c_mux_d200     │ │               │            │
│    │mc10136_pack   │ │               │            │
│    │               │ │               │            │
│    │               │ │               │            │
│    │               │ │        ( map )│            │
│    │               │ │               │            │
│    └───────────────┘ └───────────────┘(cancel)    │
└─────────────────────────────────────────────────┘
```
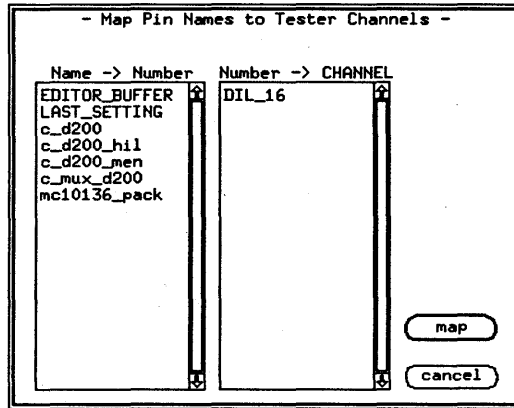
Figure 3-9. Merging Configurations - Dialog Box

The dialog box has two browsers. The one
on the left maps DUT pin names to DUT pin
numbers and the mapping files listed are stored in
/users/logname/device_name/configuration.
The files are for device mapping. The one
on the right maps DUT pin numbers to I/O
channel numbers and the files listed are stored in
/hp82000/pws/dut_boards/configuration. The files
are for DUT Board mapping.

Click the name of the template file from either or both
browsers that you want to merge with the existing
configuration. The device pin numbers and tester
channel allocation will then be taken over from the file(s)
that you specified.

Inserting Lines into an Existing Configuration

If you wish to insert lines into an existing configuration, you need to select the get channels option on the Select pulldown menu. A dialog box is displayed as shown in Figure 3-10.

```
┌──────────────────────────┐
│  - get channels -        │
│                          │
│  no. of lines ▓▒▒▒       │
│                          │
│  ( get )  ( cancel )     │
└──────────────────────────┘
```
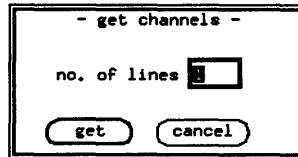
Figure 3-10.
Inserting Tester Channels - Dialog Box

Type in the number of extra channels that you want inserted. The system automatically searches for spare channels and allocates them. The remainder of the inserted lines remain blank and you need to fill them in accordingly.

This procedure is necessary to avoid channel sharing conflicts that might arise.

Checking Your Channel Configuration

You can check the integrity of the channel configuration by selecting the check TESTER CHANNELS option on the Select pulldown menu. The procedure involves you positioning the entry field in the row of the pin you want to check and contacting the pin with the calibration probe, similar to what you've already done in Auto-mapping above. The system checks the channel's configuration and beeps and gives an error message if the probed channel does not match the definition in the Pin Configuration Window.

Defining Groups of Pins

The HP 82000 offers the possibility of grouping pins together and handling these groups in the same way that you handle individual pins. This is useful for handling address busses for example, where you are interested in the value of an address rather than its individual pins.

Defining groups can also improve the readability of the setup displays and allow you to keep an overview of the DUT data. A group of four pins for example, occupies only one column in a setup window when displayed in hexadecimal format.

Permissible Groups

Before you attempt to define any groups, there are some restrictions on group definition that you should be aware of. As mentioned in the chapter on Introduction to Testing, the choice of pin mode dictates how the channel circuitry will be configured for operation. This means that for group operations, some pin driver modes might be incompatible.

When you define a group in the Pin Configuration Window, you can mix all types of pins together, however, you might not be able to edit such a group in the other setup windows. With each setup window, the group homogeneity is checked before any group operation is allowed. Groups that are not homogeneous are not displayed in the affected setup windows and are shown in half tone in the respective define table format dialog box.

The conditions for homogeneity depend on which setup is being used. A group whose pins are homogeneous for level setup, might not be homogeneous in vector setup. The conditions for homogeneity within the different setups are as follows:

Vector Setup   the following conditions cause a group's pins not to be homogeneous:

■ mixed input and output pins

- pins with different operating modes mixed together

- pins which have active termination (ioh or iol type)

Timing Setup   the conditions for homogeneity are the same as for the Vector setup.

Level Setup   a group cannot contain a pin that has an active termination (ioh or iol pin), and 50, 200, and 400 mode output pins cannot be mixed with any other type.

Defining Groups

To define a group, you need to call up the Pin Group Definition dialog box by selecting define groups from the Select pulldown menu. The dialog box is then displayed as shown in Figure 3-11. The browser in this dialog box shows all groups that have been defined for the current configuration. To define a new group, click the ⌈new⌋ button, or, if you want to modify an existing group, mark the group in the browser and then click the new button.

```
                      - Pin Group Definition -
  grouplist:

  ┌──────────────────┐
  │Din            ⇑│
  │Mode             │
  │Qio              │
  │Qout             │
  │Term             │
  │inputs           │
  │outputs          │
  │terms            │
  │                 │
  │                 │
  │                 │              ( new )   ( done )
  │                 │
  │                 │
  │                 │
  │               ⇓│
  └──────────────────┘
```
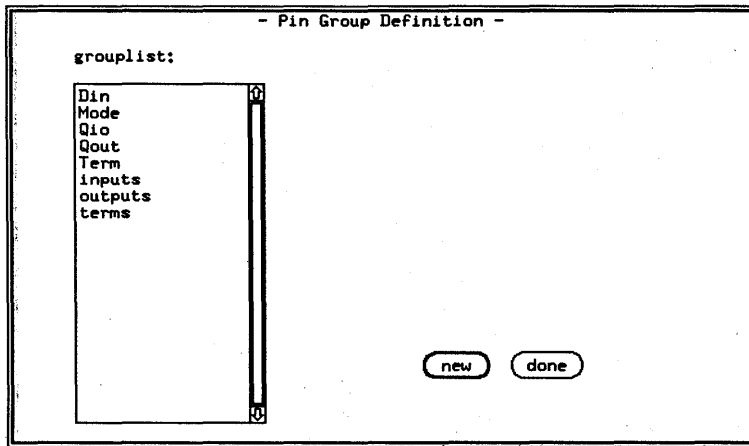
Figure 3-11. Defining Groups Dialog Box

The Pin Group Definition dialog box is then
redisplayed as shown in Figure 3-12. To define the new
group, you need to first enter the group name in the
edit box provided (under the left browser). The left
browser shows you all configured pins in the current
configuration. The right browser shows the pins that
have been defined for the group. Now mark the pins
that you want to include in the group, click the (copy)
button, followed by the (paste) button. The pins that
you selected are now displayed in the right browser, and
have been included in the group's definition.

Once you are satisfied with the group definition, click
the (save) button, which will update the current pin
configuration file.
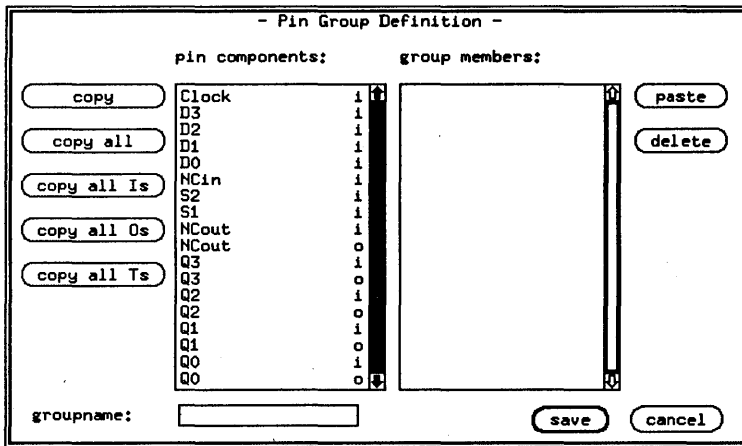
Figure 3-12. Defining New Groups

The buttons on the left side of the pin components browsers allow you to do selective copying into the group. These buttons have the following functions.

copy all	copies all pins.
copy all I	copies all input pins.
copy all O	copies all output pins.
copy all T	copies all tristate pins.

Modifying Groups To modify a group, call up the Pin Group Definition dialog box by selecting change groups from the Select pulldown menu. The dialog box is displayed similar to the one shown in Figure 3-13. Mark the group you want to change (in the example shown, the group input has been marked), and click the (change) button.

```
                    - Pin Group Definition -
  grouplist:

  ┌─────────────────────┐ ▲
  │Din                  │
  │Mode                 │
  │Qio                  │
  │Qout                 │
  │Term                 │
  │inputs               │
  │outputs              │
  │terms                │
  │                     │
  │                     │
  │                     │
  │                     │         ( change )  ( done )
  │                     │
  │                     │
  └─────────────────────┘ ▼
```
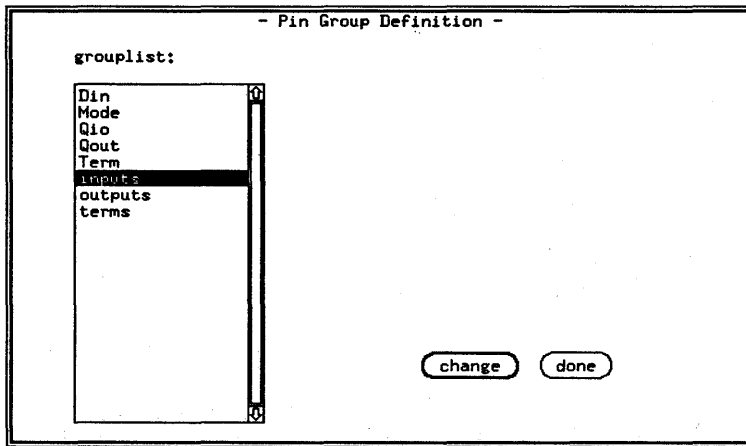
Figure 3-13. Changing Groups Dialog Box

The Pin Group Definition window is then redisplayed
as shown in Figure 3-14. The name of the group that
you selected is displayed in the lower left of the box.
As with defining groups, the left browser contains all
configured pin components, while the right browser lists
all the pin for that group. You can now add or delete
pins, in exactly the same way as you did when defining
the group. Once you have completed the changes
and are satisfied with them, click the (save) button,
otherwise you can cancel the modifications by clicking
the (cancel) button.

Figure 3-14. Changing a Group

4
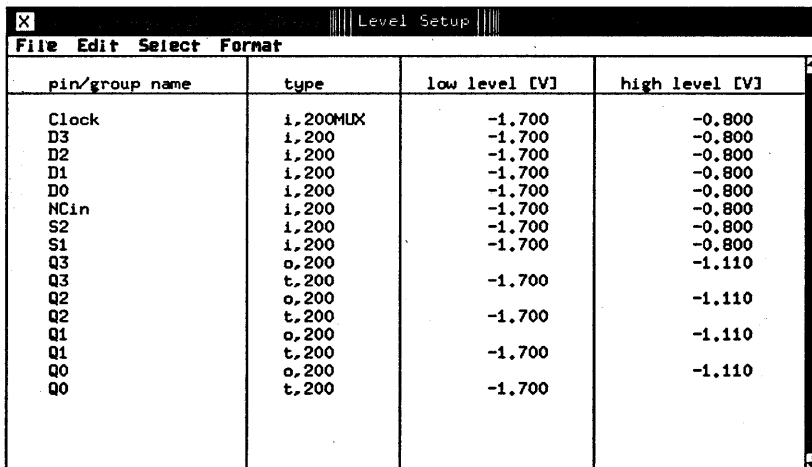
Using Level Setup

Purpose of Level Setup

The Level Setup Window allows you to define the level parameters for all the pins you have previously defined in the Pin Configuration Window.

Activating Level Setup

To activate Level Setup, click the key (Level) in the Main Menu Bar. The Level Setup Window appears on the screen, as shown in Figure 4-1.

```
[X]                        ||||Level Setup||||
File  Edit  Select  Format
                                                                    [↑]
   pin/group name        type        low level [V]      high level [V]

   Clock               i,200MUX         -1.700             -0.800
   D3                  i,200            -1.700             -0.800
   D2                  i,200            -1.700             -0.800
   D1                  i,200            -1.700             -0.800
   D0                  i,200            -1.700             -0.800
   NCin                i,200            -1.700             -0.800
   S2                  i,200            -1.700             -0.800
   S1                  i,200            -1.700             -0.800
   Q3                  o,200                               -1.110
   Q3                  t,200            -1.700
   Q2                  o,200                               -1.110
   Q2                  t,200            -1.700
   Q1                  o,200                               -1.110
   Q1                  t,200            -1.700
   Q0                  o,200                               -1.110
   Q0                  t,200            -1.700


                                                                    [↓]
```

Figure 4-1. I/O Pin Level Setup Window

The Level Setup window displays a list of the I/O pins defined in the Pin Configuration window, and lists their level definitions. Any I/O pin that has level information can be displayed here. The Device Power Supply (DPS) pins are shown in the DPS Pin Level window. The DPS pin information is accessed by clicking DPS pin level in the Select pull down menu.

Setting up the Device Power Supply

Figure 4-2 shows the DPS Pin Level setup window.

dps pin	level [V]	current [A]	off state	setup time [ms]
Vcc1	5.000	0.500	act_curr	16
Vee	-0.06	0.500	min_curr	10

Level Setup

File   Edit   Select   Format

Figure 4-2. DPS Pin Level Setup Window

The HP 82000 system can use up to eight HP 66xx series Linear System DC Power Supplies to provide DUT power. If one of these power supplies is fitted to your tester, you will be able to program the outputs in the DPS Pin Level Setup window.

The DPS Pin Level Setup window contains five columns:

- dps pin
- level [V]
- current [A]
- off state
- setup time [ms]

DPS Pin   The dps pin column shows the defined DUT power pins.
The pin names are those that you have used in the Pin
Configuration Window. You cannot edit the entries here,
they are for information only. If you haven't defined
any power pins in the Pin Configuration Window, no
DPS definitions will be displayed in the DPS Pin Level
window.

DPS Levels   The column level [V] specifies the voltage level applied
to the DPS pin. You can change the level but not the
polarity. The polarity of a power supply pin is fixed,
having been specified in the type column of the Pin
Configuration Window. If you have specified a pin type
DPS+ there, you will be able to enter positive voltages
in the level [V] column. If you have specified a pin
type DPS− there, you will be able to enter negative
voltages in the level [V] column.

Note   The level [V] column only shows the intended polarity,
not the actual polarity. The actual polarity at the DUT
is set by wiring links on the DUT Board. This procedure
is described in Chapter 2.

For the range of available voltages, as well as the
constant voltage resolutions, refer to the *Technical Data*
brochure, or to the *Operating Manual* for your DC Power
Supply.

To set the level, overtype the old value with a new one
or use the vernier to increment or decrement the level by
a small amount.

The level that you set is rounded off to the next value
programmable by the hardware. The rounded off value is
displayed after exiting the entry field.

Error Messages

Positive (or negative) real number expected.

Value out of range.

Value outside specification of DPS.

Polarity wrong.

DPS Pin Current

The column current [A] specifies the maximum current
the DPS sources or sinks. The entry is interpreted as an
absolute value and may not be negative.

For the range of available currents, as well as the
constant current resolutions, refer to the *Technical Data*
brochure, or to the *Operating Manual* for your DC Power
Supply.

To set the current, overtype the old value with a new
one or use the vernier to increment or decrement the
current by a small amount.

Note

The set current will be rounded off to the next value
programmable by the hardware. The rounded off value
will be displayed after exiting the entry field.

Error Messages

Positive real number expected.

Value out of range.

DPS Pin Off State

When you program the `disconnected` state, relays decouple DUT signal pins from test system channels. DPS connections to power pins remain.

The column `off state` specifies the programming of the current limit when the tester is in the `disconnect` state. The column offers a fill menu with two choices:

`min_curr`   programs the DPS current limit to the minimum value (close to zero).

`act_curr`   leaves the current limit at the active (`connected`) state.

Programming the current limit to go to the `act_current` setting in the `off state` allows rapid discharge of any decoupling capacitors on the DUT power pins.

To toggle the `off state` DPS current limit, click the entry field and choose the required entry from the fill menu.

Note   When the tester is in the `disconnect` state the DPS output voltage is always zero.

DPS Pin Setup Time

The column `setup time [ms]` enables you to specify the time interval between the moment the DPS settles and the test hardware goes from the `disconnect` to the `connect` state.

You can also introduce relative delays between switching on the DPS channels. This feature allows testing of devices with multiple power pins, where the order and the relative times of applying power to these pins is important.

When going from the `connect` state to the `disconnect` state, the DUT relays and DPS channels switch off more or less at the same time.

To set the time, overtype the old value with a new one or use the vernier to increment or decrement the current by a small amount.

The set time will be rounded off to the next value programmable by the hardware. The rounded off value will be displayed after exiting the entry field.

Example:

Suppose you are testing a device which requires that no signals are applied to its pins before power is present, otherwise latch-up could occur. The device needs $+12V$ (V_{cc1}), $-12V$ (V_{ee}) and $+5V$ (V_{cc2}), where the $\pm12V$ power has to be applied before the $+5V$ power.

Using a delay interval of 15 ms between applying $\pm12V$ and $+5V$, and another interval between applying $+5V$ and other signals, you would program the DPS setup time delays as in Figure 4-3.

☒	Level Setup				
File Edit Select Format					
dps pin	level [V]	current [A]	off state	setup time [ms]	
Vcc1	12.00	0.500	min_curr	25	
Vee	-12.0	0.500	min_curr	25	
Vcc2	5.000	0.500	min_curr	10	

Figure 4-3. Programming DPS Setup Time

Effects of Offline Mode Programming on DPS Settings

When no tester hardware is connected to the PWS, context dependency checking does not take place. In the case of the DPS settings it means the system accepts any value you enter.

Setting Up I/O Pin Levels

When you select the (Levels) pushbutton from the main menu, the Level Setup screen will automatically display the I/O Pin levels window. Alternatively, from the DPS Pin levels window, you can click I/O pin level in the Select pull down menu.

Level information for all pins defined in the Pin Configuration is displayed here, unless the Table Format in the Level Setup has been redefined. Defining the Table Format means selecting which pins or groups of pins will have their level information displayed in the I/O Pin Level Setup Window. This is described in the section "Selecting Level Definitions for Display ."

When you open the Level Setup Window, the level definitions show the settings currently in the tester hardware. There are always some level definitions displayed in the window. If you have just switched the HP 82000 test system on, the I/O Pin Level window will show the firmware default values or the software last used values. You can modify these values by hand, or use the file functions pull-down menu to load a level file for the device you wish to test. This will be covered in a moment in "Working with Level Files ."

If you are working in the offline mode (no tester hardware attached), the system software simulates the tester hardware and loads its own default values (or if you've saved them on last exiting the system software, the last used values).

The window displays a maximum of 20 level definitions at a time, so in the likely case that your device has more level pins than that, you will need to scroll the display. Use the scroll box inside the scroll bar on the right edge of the Level Setup Window. If you are not familiar with scroll bars see the "Common Interface Functions" chapter in the manual *Getting Started with the HP 82000.*

The table has four columns:

- pin/group name
- type
- low level [V]
- high level [V]

Pin/Group Name

The pin/group name column lists the pins or pin groups that you have selected for viewing in the Define Table Format dialog box (see "Selecting Level Definitions for Display "). Defining individual pins or pin groups for display is dealt with a little later in the pull-down menu descriptions. For the moment assume that all level pins of your device are selected for viewing. The pin names are those that you have used in the Pin Configuration definitions. You cannot edit the displayed pins from here, but you can use this column to make a pin search.

Searching for a Pin Level Definition

If your device had several hundred pins and you had to search for a particular pin by scrolling the table and scanning the display with your eyes, it would be rather tedious. To get to a level definition of a pin quickly a search function is provided.

Simply click the mouse on any pin in the column pin/group name. An entry field is created around the selected pin. Overtype the old entry with your new one and deactivate the entry field. The entry field will jump to the pin you are searching for.

This process does not destroy any data as no editing is done.

Type   This column gives the type of each pin listed in the pin/group name column. You cannot edit the type column, it is for information only.

The pin type is determined by its signal direction i or o and by the Operating Mode - that is whether the pin is clocked at 25 MHz, 50 MHz, 100 MHz, 200 MHz or 400 MHz.

Signal Direction

The signal direction refers to the DUT, not to the test system. Thus an i entry means the signal enters the DUT and stems from a machine driver channel. Similarly, an o entry describes a DUT output signal that leaves the DUT and enters a machine receiver channel.

From the pin types definable in the Pin Configuration Window the Level Setup displays only those carrying Level information. They are the following types:

i   a straight input pin
o   a straight output pin
ihs   a high speed input pin
ot   an output pin with a termination
io   a bidirectional pin
ioh   a bidirectional pin using driver high level
        termination
iol   a bidirectional pin using driver low level
        termination
nc   pin not connected

For full definitions of all pin types refer to the Pin Configuration chapter.

For the purposes of the Level Setup, all pins are split into their input and output components. An i appears in the type column of the Level Setup Window for all

pins that have an i signal component defined in the Pin Configuration Window. Similarly an o appears for those pins with an o component.

A pin defined as ot in the type column of the Pin Configuration Window is flagged with an o in the type column of the Level Setup Window, and also with a t (signifying 'termination').

A pin defined as io in the type column of the Pin Configuration Window is displayed twice in the type column of the Level Setup Window. Once as an i pin and once as an o pin.

Groups containing pins of mixed signal direction or operating mode display a question mark (?) in the type column.

Low Level

The column low level [V] gives the voltage level which defines:

for input pins	the logic "0" level
for output pins	the logic "0" compare threshold

The units are volts.

To set the level, overtype the old value with a new one or use the vernier to increment or decrement the level by a small amount.

Note

The set level will be rounded off to the next value programmable by the hardware. The rounded off value will be displayed after exiting the entry field.

Error Messages

Positive (or negative) real number expected.

Value out of range.

High Level The column high level [V] gives the voltage level which defines:

for input pins   the logic "1" level

for output pins   the logic "1" compare threshold

The units are volts.

To set the level, overtype the old value with a new one or use the vernier to increment or decrement the level by a small amount.

Note The set level will be rounded off to the next value programmable by the hardware. The rounded off value will be displayed after exiting the entry field.

Error Messages

Positive (or negative) real number expected.

Value out of range.

Single and Dual Level Compare

The hardware performs dual level edge compare in 25 mode, 50MUX mode, 100 mode, 200MUX mode, 200COM mode and 400MUX mode. In the 50 mode, 200 mode and 400 mode it performs single level compare.

Summary of Drive Levels, Ranges and Swings Table 4-1 gives a summary of the drive logic levels and drive level ranges available:

Table 4-1. Input Level Ranges

Parameter	Implicit Unit	Range (D100, D200 & D400)	Range (D50)	Range (ihs pins)	Default
logic "0" level	V	−4.000..7.800	−2.000..6.500	−4.000..4.800	−1.700
logic "1" level	V	−3.800..8.000	−2.500..7.000	−3.800..5.000	−0.800

There are swing limitations which mean that the minimum swing between logic "0" level and logic "1" level is 0.2 V and the maximum permitted swing is 8.0 V (7.0 V for the 50 MHz boards). Violations of these minimum or maximum swing values will cause an error.

When an output channel is configured with active termination (an ot type pin), the drive part of the channel is forced to a static low level. The single level is programmed in the low level [V] column. You cannot program the high level [V] column; anything you enter in this column is ignored.

Errors

Errors will be generated for:

- low level out of range
- high level out of range
- swing violation

Warnings

Warnings will be generated if:

- Auto correction occurs
- Level programming exceeds the hardware specs

Summary of Receive Levels, Ranges and Swings

Table 4-2 and Table 4-3 give summaries of receive levels available for all operating modes:

Table 4-2. Output Logic Levels

Operating Mode	Low Level	High Level
50, 200, 400	not programmable	threshold for single level compare
25, 50MUX, 100, 200MUX, 200COM, 400MUX	lower threshold for dual level compare	upper threshold for dual level compare

Table 4-3. Output Threshold Ranges

Parameter	Implicit Unit	Range (D100,D200,D400)	Range (D50)	Default Value
logic 0 threshold	V	−4.000..7.994	−2.000..6.950	−1.470
logic 1 threshold	V	−3.994..8.000	−2.010..7.000	−1.110

As in the case of driver level programming, there is a swing limitation such that there must be a minimum swing between logic "0" threshold and logic "1" threshold. The minimum swing limitation depends on the defined pin's attributes. The minimum threshold swing is 0.006 V (0.010 V for 50 MHz I/O Boards).

In the case of a channel operating in 200 mode, the required single threshold is set by the high level [V] threshold. The low level [V] threshold is ignored and is auto-corrected to a value dependent on the given threshold.

Table 4-4 shows the termination of ot pins.

Table 4-4. Termination of ot Pins

Operating Mode	Low Level	High Level
All Modes	termination level	not programmable

Errors

Errors will be generated for:

- low level out of range
- high level out of range

Warnings

A warning will be generated if:

- auto-correction has occurred
- level programming exceeds the hardware specifications

Autocorrection

If you program a level to a value which violates the maximum or minimum swing constraints, the system will autocorrect as follows:

- updating the low level beyond the maximum swing value causes the high level to autocorrect to the maximum swing.

- updating the high level beyond the maximum swing value causes the low level to autocorrect to the maximum swing.

- updating the low level beyond the minimum swing value causes the high level to autocorrect to the minimum swing.

- updating the high level beyond the minimum swing value causes the low level to autocorrect to the minimum swing.

DC Loading

DC loading has uses for testing devices employing bidirectional signal pins and tristate. These are for example TTL low current devices or CMOS devices.

The model D50 has built in DC loading circuitry. The circuitry consists of pull-up and pull-down resistors and associated switches that can be programmed to pull the output of the pin driver in the appropriate direction. This is shown in Figure 4-4.
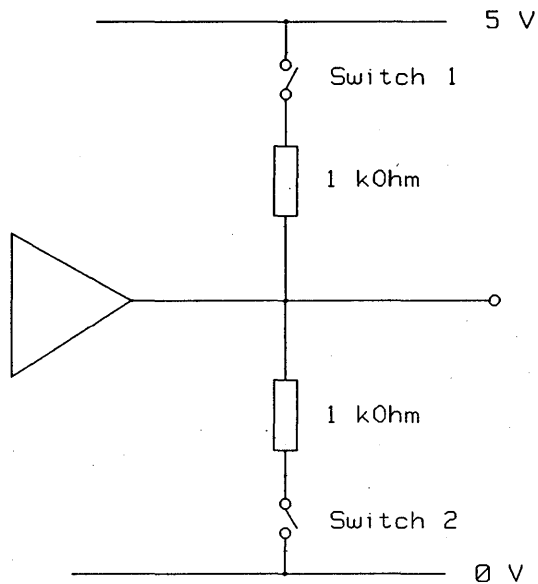
Figure 4-4. DC Loading Circuitry

Table 4-5 shows the programming options and the associated loading conditions.

Table 4-5. DC Loading Programming Options

Switch 1	Switch 2	DC Loading
open	open	none
closed	open	5V_1k
open	closed	0V_1k
closed	closed	2.5V_500

An extra column in the Level Setup Window of the D50 system allows DC loading conditions to be programmed, as shown in Figure 4-5. Note how it differs from the

Level Setup window for D100, D200 and D400 systems,
shown in Figure 4-1.

pin/group name	type	low level [V]	high level [V]	dc loading
Clock	i.50MUX	-1.700	-0.800	none
D3	i.50	-1.700	-0.800	none
D2	i.50	-1.700	-0.800	none
D1	i.50	-1.700	-0.800	none
D0	i.50	-1.700	-0.800	none
NCin	i.50	-1.700	-0.800	none
S2	i.50	-1.700	-0.800	none
S1	i.50	-1.700	-0.800	none
Q3	o.50		-1.110	none
Q3	t.50	-1.700		none
Q2	o.50		-1.110	none
Q2	t.50	-1.700		none
Q1	o.50		-1.110	none
Q1	t.50	-1.700		none
Q0	o.50		-1.110	none
Q0	t.50	-1.700		none

Figure 4-5. Level Setup Window of D50 Tester

Working with Level Files

The Menu Bar at the top of the Level Setup Window gives a number of options. The first option is (File). Clicking the mouse on (File) opens a pull down menu, listing a number of further options.

Note

The pull down menu options and their associated dialog boxes that are common to most setup screens, for example the load option and the load file dialog box, are described in the reference chapter on 'Common Functions'.

File Options   load

Calls up the **load file** dialog box where the Level file to be loaded is specified. Select with the mouse from the file browser or enter required filename in the edit field. The files are contained in the **levels** subdirectory of the current device directory. The full pathname to such a directory could be for example: /users/cae/mc10136/levels.

If you wish to load files from a different directory, you need to enter the new pathname into the device pathname entry field of the dialog box. This is not a full pathname but a "device pathname". The System Software automatically supplies the last part of the pathname, in this case **levels**.

Example: To load files from say /users/donald/mc10131/levels, you would enter in the device pathname entry field: /users/donald/mc10131.

A hotkey ^L is provided to get to the load file dialog box quickly. Just press the (CTRL) (L) keys simultaneously.

catalog
print

prints a listing of all Level files in the current levels subdirectory.

copy

copies one or more Level files from a device **levels** subdirectory to the current **levels** subdirectory. You have to supply the device pathname of the directory to copy from in the entry field. Any files found in this directory will be listed in the copy file browser. Select the file(s) to copy with the mouse.

delete	deletes the Level file(s) from the current **levels** subdirectory that you mark with the mouse in the delete file browser.
save	allows you to save a Level setup to disk under its original name. The setup will be saved to the current **levels** subdirectory. If you have not specifically loaded a Level file using the load function you will get the Save_as dialog box with an empty entry field. You will have to supply a filename to save to.
save_as	allows you to save a Level Setup to disk under a filename of your choice. The setup will be saved to the current **levels** subdirectory.

Editing Functions

The menu bar option (Edit) enables you to edit data in the I/O Pin Level window.

copy	allows you to copy data from an activated entry field in an editable column into the paste buffer. Use the mouse and the **Edit** pull down menu, or the ^C hotkey sequence by entering (CTRL) (C).
paste	allows you to paste the contents of the paste buffer into an activated entry field in an editable column. Use the mouse and the **Edit** pull down menu, or the ^P hotkey sequence by entering (CTRL) (P).

Selecting Level Definitions for Display

The menu bar option (Format) allows you to define the format of the I/O Pin Level window. Through the Define table format dialog box you can select Level definitions of those device pins or groups of device pins, that you wish to see in the I/O Pin Level Setup Window (see Figure 4-1).

If you haven't selected any pins in the Define table format dialog box, the I/O Pin Level window shows the level definitions of all pins that have been defined in the Pin Configuration Window.

If you have selected some pins or pin groups in the Define table format dialog box, then the Level definitions of those pins or pin groups will be displayed in the window.

Selecting All Configured Level Definitions

The dialog box Define table format contains three browsers. The browser table format contains a list of all entries, be it pins or pin groups, currently selected for display in the I/O Pin Level window.

It also gives the type of entry:

i   input pin
o   output pin
t   termination voltage for ot pin
g   group (of input or output pins)

As the browser can display only 15 entries at a time you have to use the scroll bar to see them all.

On the right of this browser there are the pushbuttons (set default) and (clear all), see Figure 4-6.

```
                     - Define table format -
pin
components:  ┌─────────────────┐     table format:
            │Clock        i│▲│     ┌─────────────────┐          ┌──────────┐
 ┌──────┐   │D3           i│ │     │Clock       i │▲│          │  paste   │
 │ copy │   │D2           i│ │     │D3          i │ │          └──────────┘
 └──────┘   │D1           i│ │     │D2          i │ │
            │DO           i│ │     │D1          i │ │          ┌──────────┐
            │NCin         i│ │     │DO          i │ │          │ delete   │
            │S2           i│ │     │NCin        i │ │          └──────────┘
            │S1           i│ │     │S2          i │ │          ──────────────
            │NCout        i│ │     │S1          i │ │          ┌─────────────┐
            │NCout        o│▼│     │NCout       i │ │          │ set default │
            └─────────────────┘     │NCout       o │ │          └─────────────┘
                                    │Q3          i │ │
grouplist:  ┌─────────────────┐     │Q3          o │ │          ┌─────────────┐
            │All_in       │▲│     │Q2          i │ │          │ clear all   │
            │All_out      │ │     │Q2          o │ │          └─────────────┘
 ┌──────┐   │Din          │ │     │Q1          i │▼│
 │ copy │   │Mode         │ │     └─────────────────┘
 └──────┘   │Qout         │ │
            │             │ │                           ┌──────┐   ┌────────┐
            │             │ │                           │ save │   │ cancel │
            │             │▼│                           └──────┘   └────────┘
            └─────────────────┘
```
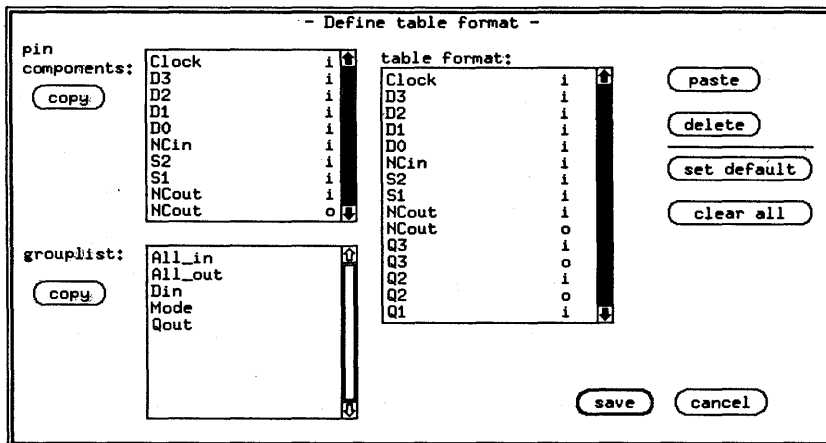
Figure 4-6. Define Table Format Dialog Box

To display the Level definitions of all configured Level
pins, click the mouse on the (set default) pushbutton. The
names of all configured pins also appear in the table
format browser. Alternatively, you can click the mouse
on the (clear all) pushbutton. This also displays the Level
definitions of all configured Level pins, but the table
format browser will be empty.

These two conditions are the starting points for editing
the entries in the table format browser. You can now
delete some pins or pin groups from the table format
browser or paste them to the browser.

Deleting Pins

To select a pin or a pin group for deletion from the table
format browser, click it with the mouse. The selected
entry will appear in inverse video. To select several
adjacent entries, click the mouse on the first entry you
wish to delete and without releasing the mouse button
draw the mouse across all the entries to be deleted. The
selected entries will appear in inverse video. To select
several non-adjacent entries, hold down the (Shift) key

and click the required entries with the mouse. The selected entries will appear in inverse video.

To delete the marked entries, click the (delete) pushbutton. The deleted entries are moved to the paste buffer. If you've made a mistake, you can get them back by clicking the (paste) pushbutton.

Adding Pins

Adding pins to the table format browser entails two steps:

1. copying pins or pin groups to the paste buffer (described in the next two sections)
2. pasting the buffer into the table format browser

Pasting into the Table Format Browser

The paste buffer contains pins or pin groups you have last copied into it. To append the contents of the paste buffer to the contents of the table format browser, just click the (paste) pushbutton.

To insert the contents of the paste buffer into the table format browser, select an entry in the browser where you want the paste buffer copied. The selected entry will be displayed in inverse video. Now click the (paste) pushbutton. The paste buffer will be inserted before the selected entry.

Selecting Level Definitions of Pins

The browser pin components contains all DUT pins that have been defined in the Pin Configuration Window and carry Level information. It includes all pins except DPS+, DPS− and dc pins.

To copy one or more pins to the paste buffer, select the pin(s) with the mouse and click the (copy) pushbutton. Use the mouse the same way as before.

Selecting Level Definitions of Pin Groups

The browser `grouplist` contains all groups of DUT pins that have been defined in the Pin Configuration Window and carry Level information.

To copy a group or several groups into the paste buffer follow the same procedure as for pin components.

Restrictions

- The active termination (t) component of a pin may not be mixed with any other pins or pin components;

- o pins with single-level edge compare (50, 200 and 400 modes) may not be mixed with pins of a different type.

Example - Defining the Table Format:

Figure 4-7 shows the `table format` browser contains the input pins S2, S1 and the output pins Q3 to Q0. In order to add the pins D3 to D0 to the existing display in the I/O Pin Level window, such that they appear between pins S2, S1 and pins Q3 - Q0, you have to perform the following steps:

1. Mark the pins in the browser `pin components`, that you wish to display in the I/O Pin Level window - in this case the pins D3 to D0.
2. Mark the pin in the browser `table format` before which you wish to insert the extra pins - in this case Q3.
3. Click the mouse on the pushbutton (copy) next to the `pin components` browser to copy the marked pins into the paste buffer.
4. Click the mouse on the pushbutton (paste) to paste the buffer into the `table format` browser at the selected place.

```
                        - Define table format -
pin
components:  Clock           i ▲   table format:
 ( copy )    D3             i      S2             i  ▲    ( paste )
             D2             i      S1             i
             D1             i      D3              o      ( delete )
             D0             i      Q2              o
             NCin           i      Q1              o    ( set default )
             S2             i      Q0              o
             S1             i                           ( clear all )
             NCout          i
             NCout          o ▼

grouplist:   Din            ▲
 ( copy )    Mode
             Qio
             Qout
             Term
             inputs
             outputs                                 ( save )  ( cancel )
```
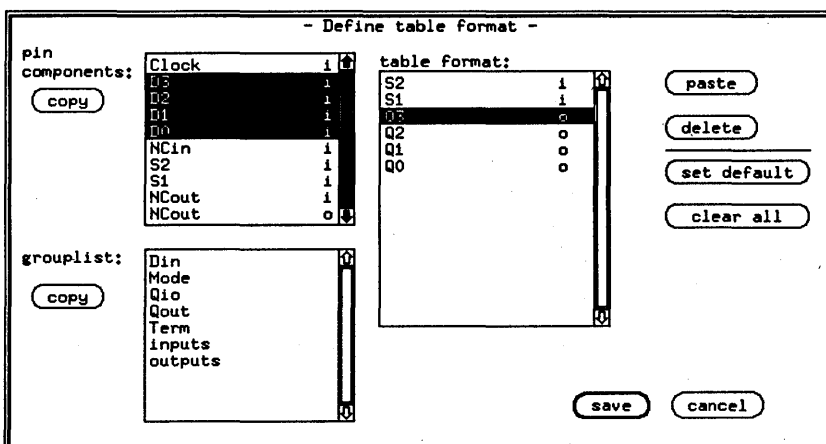
Figure 4-7. Example - Defining the Table Format

Configuring a Pin Group

Any pin groups listed in the browser grouplist have
been made up in the Pin Configuration Window. Once
created there, they cannot be changed or edited in the
Define table format dialog box. They can only be
added to or deleted from the display of the I/O Pin
Level window. If you wish to modify a pin group, you
have to do it in the Pin Configuration Window.

The pin groups listed in the browser grouplist normally
contain Level definitions of similar pins. For instance,
all input data pins of a device could be grouped under
the name D_{in}. If all these pins have the same Level
characteristics, they can be combined into one group and
the group then displayed in the I/O Pin Level window,
rather than displaying the pins individually. This is the
function of pin groups - purely to help you edit the Level
definitions. It is particularly useful when testing high
pin-count devices.

Interactions between Pins and Pin Groups

Grouping similar pins together does not mean the individual pins in the group can no longer be displayed in the I/O Pin Level window. Both the individual pins and the pin groups made up of these pins can be displayed at the same time.

If you do this, you have to exercise a little care when modifying any of the Level parameters in the I/O Pin Level window.

Changing a Group Level Definition . If you change the Level definitions of any of the groups that you have defined in the table format, all pin entries in the affected column relevant to the modified group will change to the new setting.

Changing a Pin Level Definition. The level definition of a group is defined only if the level definitions of all pins making up the group are consistent.

If you change the Level definitions of any of the pins contained in a group, the Level definition of the group is no longer consistent. This is signified by a question mark (?) displayed in the affected column against the group containing the changed pin. This is because the group definition no longer agrees with the individual pin definitions in the group. The hardware will still function in this condition, since it is the individual pin definitions that determine the state of the hardware, not the group definitions.

Multiple Pin Definitions. If you change the Level definitions of a pin that has been defined more than once in the table format, the rest of the definitions of that pin will also be changed. Remember, there is only one hardware resource per pin.

Use of a Pin in Several Pin Groups. The use of one pin in several groups is possible. However bear in mind that the Level definitions of these groups will be consistent only if the level definitions of all the pins contained in the groups are equal.

If you change the definition of one of the groups, the rest of the groups will have inconsistent definitions and will display question marks (?) in the columns affected.

Note

When creating, deleting and manipulating pin groups, remember that a group includes not just the level definitions for all its pins, but the timing and vector definitions as well.

Effects of Group Changes on Table Format Definition

Changing a group in the Pin Configuration Window such that it contains both input and output components of channels renders the group unusable in Level Setup. The group's name is displayed in the grouplist in *shadow font* and a star appears in the table format next to the group's name.

If you delete a group in the Pin Configuration Window which you have been using in the Level Setup, the system software removes the group from the I/O pin level window, as well as from the grouplist. The group's name is not deleted from the table format, but a star appears next to the groups name.

Timing Reference Modes

The Select pull down menu allows you to toggle between 50% Reference and Freeze Reference.

50% Reference Mode

In this mode, the HP 82000 sets the signal edge timing reference to the *mid point* of the swing between low level and high level. This condition is automatically maintained by adding a timing correction at the pin driver.

Freeze Reference Mode

When this mode is chosen, the automatic correction facility is disabled and the last correction value is used. This allows signal levels to be changed without affecting the timing.

For a full explanation and details of the application of this facility, refer to Appendix B.

5

Using Timing Setup

Purpose of Timing Setup

The Timing Setup window allows you to define the timing parameters for the pins you have previously defined in the Pin Configuration Window.

Activating Timing Setup

To activate the Timing Setup window, click the key Timing in the Main Menu.

As Figure 5-1 shows, the window is divided into two areas: an upper area containing the global settings; and a lower area containing a scrollable table of timing definitions. Each line in the table represents a timing definition for one device pin (or a group of pins). A detailed description of timing definition follows later.

The Global Settings area shows you the current settings of the clock period, clock frequency and the clock source. You can change the entries for the period and the clock source. You cannot change the entry for the frequency directly, it is calculated from the period.

```
┌─────────────────────────────────────────────────────────────────────────┐
│ ☒                          ║║║ Timing Setup ║║║                         ▲ │
│ File  Edit  Select  Format                                                │
│                                                                           │
│      period          │ 20.0 │ ns        frequency(100MHz pins)  50.00  MHz│
│                                                                           │
│      clock source  │intern│                                               │
├───────────────────┬──────────┬──────────┬─────────────────┬──────────────┤
│  pin/group name   │   type   │  format  │ leading edge[ns] │trailing edge[ns]│
├───────────────────┼──────────┼──────────┼─────────────────┼──────────────┤
│ ▓lock             │  i,100   │  RZ      │      9.00       │    19.00     │
│ Mode              │  i,100   │  DNRZ    │      2.00       │     2.00     │
│ NCin              │  i,100   │  DNRZ    │      2.00       │     2.00     │
│ Din               │  i,100   │  RZ      │      2.00       │    15.00     │
│ Qout              │  o,100   │  edge    │     18.00       │              │
│ NCout             │  o,100   │  edge    │     18.00       │              │
│ S2                │  i,100   │  DNRZ    │      2.00       │     2.00     │
│ S1                │  i,100   │  DNRZ    │      2.00       │     2.00     │
│ D3                │  i,100   │  RZ      │      2.00       │    15.00     │
│ D2                │  i,100   │  RZ      │      2.00       │    15.00     │
│ D1                │  i,100   │  RZ      │      2.00       │    15.00     │
│ Q3                │  o,100   │  edge    │     18.00       │              │
│ Q2                │  o,100   │  edge    │     18.00       │              │
│ Q1                │  o,100   │  edge    │     18.00       │              │
│ Q0                │  o,100   │  edge    │     18.00       │            ▼ │
└───────────────────┴──────────┴──────────┴─────────────────┴──────────────┘
```

Figure 5-1. Timing Setup Main Window

Setting the Clock Period

The clock period determines the test vector rate. When you open the Timing Setup window a period will be displayed. This will either be the default period or the last used period in the LAST_SETTING file.

To set the clock period, click the period entry field and type in the new value. The value must be a positive, real, decimal number. The units are always nanoseconds (ns). When you have set the new period, exit the entry field. This downloads the new value into the hardware.

Alternatively you can use the vernier to increment or decrement the period by a small amount.

Note

The set period will be rounded to the next value programmable by the hardware. The rounded value will be displayed after exiting the entry field.

A dependency exists between the programmed period, and the leading and trailing edge delay. This is discussed later in this chapter.

Clock Period Restrictions

The range of values allowed for the Clock Period depends on the type of tester hardware (D50, D100, D200 or D400), and on the operating modes that you have selected in the Pin Configuration window.

The possible range of values for the period is as follows:

D50	25 mode	40.0 ns to 99900 ns
	50 mode	20.0 ns to 49950 ns
D100	100 mode	9.0 ns to 99900 ns
D200	100 mode	9.0 ns to 99900 ns
	200 mode	4.5 ns to 49950 ns
	200MUX mode	4.5 ns to 49950 ns
D400	200COM mode	4.5 ns to 49950 ns
	400 mode	2.25 ns to 24975 ns
	400MUX mode	2.25 ns to 24975 ns

With D50, D200 and D400 systems, the value that you enter in the period field sets the clock period of the *highest frequency pins that you are using.* The period of the slower pins is *two* or *four* times this value. The current frequencies are always displayed in the top right corner of the Timing Setup window.

Effects of Offline Mode Programming on the Clock Period

When the tester hardware is connected, the value you enter in the period field is rounded up to or down to the nearest value that the hardware can implement. The new value is displayed when you download your values to the hardware.

When no tester hardware is connected to the PWS, the context-dependency check does not take place. In the case of the clock period it means that you can enter any value you like, without it being checked for validity. Thus you could type 1.0 (ns) into the period field and the software promptly obliges by displaying a frequency of 1000 MHz!

Setting the Clock Source

As well as running the HP 82000 from the built-in PLL clock circuitry, you can also synchronize the internal clock to an external clock source. This is discussed in the section "Setting up External Clock Synchronization ."

To toggle the clock source pick the clock source entry field and choose intern or extern from the fill menu.

You can also type the words extern or intern directly into the activated entry field.

You can find further information about adjusting the clock synchronization in Application Note 390, *Synchronizing the HP 82000 to External Equipment.*

Timing Definition Layout

The lower area in the Timing Setup Window contains a table of timing definitions. Any pin that has timing information associated can be displayed here. You cannot display for example power pins as they carry no timing information.

Timing information for all pins defined in the Pin Configuration is displayed here, unless the Table Format in the Timing Setup has been redefined. Defining the Table Format means defining which pins or groups of pins will have their timing information displayed in the lower part of the Timing Setup Window. This is discussed in the section "Selecting Timing Definitions for Display ."

When you open the Timing Setup Window, the timing definitions show the settings currently in the tester hardware. There are always some timing definitions displayed in the table. If you have just switched the HP 82000 test system on, the table of timing definitions will show the firmware default values or the software last

used values. You can modify these values by hand, or use the file functions pull-down menu to load a timing file for the device you wish to test. This is described in "Working with Timing Files ."

The table has five columns:

- `pin/group name`
- `type`
- `format`
- `leading edge[ns]`
- `trailing edge[ns]`

Pin/Group Name

The pin/group name column lists the pins or pin groups that you have selected for viewing in the Define Table Format dialog box (see "Selecting Timing Definitions for Display "). Defining individual pins or pin groups for display is dealt with a little later in the pull-down menu descriptions. For the moment assume that all timing pins of your device are selected for viewing. The pin names are those that you have used in the Pin Configuration definitions. You cannot edit the displayed pins from here, but you can use this column to make a pin search.

Searching for a Pin Timing Definition

If your device had several hundred pins and you had to search for a particular pin by scrolling the table and scanning the display with your eyes, it would be rather tedious. To get to a timing definition of a pin quickly a search function is provided.

Simply click the mouse on any pin in the column `pin/group name`. An entry field is created around the selected pin. Overtype the old entry with your new one and deactivate the entry field. The entry field will jump to the pin you are searching for.

This process does not destroy any data as no editing is done.

Type This column gives the type of each pin listed in the pin/group name column. You cannot edit the type, it is for information only, coming from the Pin Configuration Window.

The pin type is determined by its signal direction i or o and by the Operating Mode.

Signal Direction

Of the pin types definable in the Pin Configuration window, the Timing Setup only displays those carrying timing information. They are the following types:

i   input pin
ihs   high speed input pin
o   output pin
ot   output pin with active termination
io   bidirectional input/output pin
ioh   bidirectional pin using the driver low level as
        termination
iol   bidirectional pin using the driver high level as
        termination
nc   pin not connected

For full definitions of all pin types refer to the Pin Configuration chapter.

For the purposes of the Timing Setup all pins are split into their input and output components. An i appears in the type column of the Timing Setup Window for all pins that have an i signal component defined in the Pin Configuration Window. Similarly an o appears for those pins with an o component.

Thus for example a pin defined as ot in the type column of the Pin Configuration Window is flagged with an o in the type column of the Timing Setup Window.

A pin defined as io in the type column of the Pin Configuration Window is displayed twice in the type column of the Timing Setup Window. Once as an i pin and once as an o pin.

Format The column format gives the Drive Format of the stimulus data going to the DUT and the type of Receive Format used to evaluate the data returning from the DUT. The drive format is denoted in the type column with an i, the Receive Format is denoted with an o. To change the format just click the current format of the required channel and use the fill menu, alternatively type the format you require directly over the old entry.

Drive Formats

The HP 82000 offers you the following drive formats:

DNRZ   Delayed Non-Return to Zero.

         In some modes, tristate (DNRZ+High-Z) is also available in this mode. If you define a trailing edge in this mode, the format will DNRZ and then tristate after the time defined by the trailing edge.

R1   Return to One

RC   Return to Complement

RZ   Return to Zero

RI   Return to Inhibit

Chapter 3 lists the driver formats available for the pins in each operating mode. The fill menu in the format column is automatically adjusted by the software, so that it only offers you valid drive formats for the particular pin type and operating mode that you are using.

Receive Formats

You can use the following receive formats:

edge   edge compare

window   window compare

Chapter 3 lists the receiver formats available for the pins in each operating mode. The fill menu in the format column is automatically adjusted by the software, so that it only offers you valid formats for the particular pin type and operating mode that you are using.

Compare Modes

Table 5-1 gives a summary of compare formats available. When an edge occurs, the DUT output data is sampled when you use the Acquisition Mode and compared (against expected data) when you use the Real Time Compare Mode. This is the case in Edge Compare.

In Window Compare the DUT output data is sampled throughout the duration of the time window in the Acquisition Mode and compared (against expected data) in the Real Time Compare Mode.

Table 5-1. Receive Formats and Compare Modes

Operating Mode	Receive Format	Leading Edge (LE)	Trailing Edge (TE)
25 100	Edge=default Window	Sample/Compare edge Sample/Compare begin	Not programmable Sample/Compare end
50, 200 200COM, 400MUX 400	Edge	Sample/Compare edge	Not programmable
50MUX 200MUX	Edge=default Window	Sample/Compare edge Sample/Compare begin	Not programmable Sample/Compare end

Leading Edge   This column gives the delay of the leading edge, relative
to the beginning of the cycle.

For input pins (drive channels), this delay defines the
first signal transition (if programmed) of the stimulus
data in a cycle.

For output pins (receive channels), this delay defines
the sampling point in the edge compare mode, or the
beginning of the *compare window* in the window compare
mode.

Cycle Boundary Crossing

In certain conditions, the leading edge delay can be
extended *outside* the cycle period. Table 5-2 shows the
range of delay available.

Note

Most of the AC Tests will not work when the leading
edge delay crosses a cycle boundary: only the functional
ac test and the acquire data test will still operate.

Table 5-2. Leading Edge Delays

Operating Mode	Period	Maximum Leading Edge Delay
25 or 50	40 ns to 41.7 ns 41.7 ns to 181.3 ns 181.3 ns to 600 ns over 600 ns	1 cycle period (4 cycle periods)−125 ns 600 ns 1 cycle period
50MUX	40 ns to 50 ns 50 ns to 207.1 ns 207.1 ns to 600 ns over 600 ns	1 cycle period (3.5 cycle periods)−125 ns 600 ns 1 cycle period
100 or 200	10 ns to 24.7 ns 24.7 ns to 168.5 ns 168.5 ns to 600 ns over 600 ns	1 cycle period (4 cycle periods)−74 ns 600 ns 1 cycle period
200MUX	10 ns to 29.6 ns 29.6 ns to 192.6 ns 192.6 ns to 600 ns over 600 ns	1 cycle period (3.5 cycle periods)−74 ns 600 ns 1 cycle period
200COM, 400 or 400MUX	10 ns to 27.6 ns 27.6 ns to 190.6 ns 190.6 ns to 600 ns over 600 ns	0.5 cycle period (3.5 cycle periods)−76 ns 600 ns 1 cycle period

The values given in Table 5-2 give the *minimum* guaranteed range of values accepted by the system before an error message is issued in the Report window. In certain circumstances, the system will accept a leading edge delay greater than these values.

Trailing Edge This column gives the delay of the trailing edge, referenced to the beginning of the cycle. For input pins this delay defines the last signal transition (if programmed) of the stimulus data in a cycle. For output pins this delay defines the end of the compare window in the window compare mode. Table 5-3 and Table 5-4 show the limits that you can set for the trailing edges of input and output pins.

Table 5-3. Trailing Edge Delays—Input Pins

Trailing Edge Limits	D50	D100, D200 and D400 (except ihs pins)	ihs type pins (see note)
Minimum Delay	LE + 3 ns	LE + 1 ns	LE + 1 ns
Maximum Delay	LE + (cycle - 3 ns)	LE + (cycle - 1 ns)	LE + (cycle - 9.95 ns)

(where: LE = Leading Edge delay)

Note   You can further reduce the pulse width produced by ihs pins, by entering a trailing edge value of 0.5 ns. However, the tester is then outside its calibrated operating range and the true width of this pulse may be nearer 700 ps.

Table 5-4. Trailing Edge Delays—Output Pins

Trailing Edge Limits	D50	D100 and D200	D400
Minimum Delay	LE + 3 ns	LE + 2.5 ns	N/A
Maximum Delay	LE + (cycle - 3 ns)	LE + (cycle - 2.5 ns)	N/A

(where: LE = Leading Edge delay)

Autocorrection

If you reduce the period such that it is no longer compatible with the current leading and trailing edge delays, the system will check back to confirm whether you really wish to program the new value. If yes, the system will autocorrect the edge delays to maximum values in order to fit in with the new period.

If you move one of a pair of edges to a value which violates the maximum time difference between the leading edge and the trailing edge, (for the programmed period) the system will issue a warning and autocorrect by setting the other edge to the maximum time difference from the moved edge.

Similarly, if you move one of the edges to a value which violates the minimum time difference between the leading edge and the trailing edge, the system will set the other edge to the minimum time difference from the moved edge.

In addition, the system may autocorrect if you change the drive format from DNRZ to Rx or vice versa. No autocorrection takes place when changing the receive format from edge compare to window compare or vice versa. For numerical values refer to the Technical Specifications Data Sheet.

Example: You are running at a period of 10ns using the RZ format. The leading edge is set to 4ns, the trailing edge to 13ns. You now change the format to DNRZ. The system autocorrects by moving the trailing edge to 10ns.

Note

Autocorrection does not operate when the system is offline (with the tester hardware disconnected).

Working with Timing Files

The Menu Bar at the top of the Timing Setup Window gives a number of options. The first option is (File). Clicking the mouse on (File) opens a pull down menu, listing a number of further options.

File Options

load

calls up the load file dialog box where the timing file to be loaded is specified. Select with the mouse from the file browser or enter required filename in the edit field. The files are contained in the timing subdirectory of the current device directory. The full pathname to such a directory could be for example: /users/cae/mc10136/timing.

If you wish to load files from a different directory, you need to enter the new pathname into the device pathname entry field of the dialog box. This is not a full pathname but a "device pathname". The System Software automatically supplies the last part of the pathname, in this case timing.

Example: To load files from say /users/donald/mc10136/timing, you would enter in the device pathname entry field: /users/donald/mc10136.

A hotkey ^L is provided to get to the load file dialog box quickly. Just press the (CTRL) (L) keys.

catalog print

prints a listing of all timing files in the current timing subdirectory.

copy

copies one or more timing files from a device timing subdirectory to the current timing subdirectory. You

have to supply the device pathname of the directory to
copy from in the entry field. Any files found in this
directory will be listed in the copy file browser. Select
the file(s) to copy with the mouse.

delete

deletes the timing file(s) that you mark with the mouse
in the delete file browser from the current timing
subdirectory.

save

allows you to save a timing setup to disk under its
original name. The setup will be saved to the current
timing subdirectory. If you have not specifically loaded
a timing file using the load function you will get the
Save_as dialog box with an empty entry field. You will
have to supply a filename to save to.

save_as

allows you to save a timing setup to disk under a
filename of your choice. The setup will be saved to the
current timing subdirectory.

Editing Functions

The menu bar option (Edit) enables you to edit data in
the table of timing definitions.

copy   allows you to copy data from an activated
              entry field in an editable column into the
              paste buffer. Use the mouse and the **Edit**
              pull down menu, or the ^C hotkey sequence
              by entering (CTRL) (C).

paste   allows you to paste the contents of the
              paste buffer into an activated entry field in
              an editable column. Use the mouse and

the **Edit** pull down menu, or the ^P hotkey
sequence by entering (CTRL) (P).

Setting up External Clock Synchronization

The Menu Bar option (Select) allows setting up the
synchronization for the external clock. The tester
hardware can be frequency synchronized to a multiple
or a submultiple of an external clock signal applied at
the External Clock input. In addition, it can also be
phase synchronized to an external synchronization signal
applied at the Sync input.

You can set up the external clock and the sync clock
only if the clock source entry in the global settings
area of the Timing Setup screen (see Figure 5-1 and the
section "Setting the Clock Source ") is set to extern.

Frequency Synchronization

The test system hardware uses phase-locked-loop (PLL)
techniques to generate its internal clock. An External
Clock input is provided to enable you to connect an
external clock source and this clock source then becomes
the timing reference for the test system. The internal
clock still generates the clock signal for the test system,
but the signal is synchronized in frequency to the
external clock, so in order to generate an accurate, stable
timing signal, you must enter the value of your external
frequency as precisely as possible in the Setup window.

The external clock input circuits are designed to accept a
wide range of frequencies (and voltage swings) into 50 Ω:

Application Note 390, *Synchronizing the HP 82000 to
External Equipment,* gives further details about how to
syncronize the HP 82000 Master Clock circuit.

ext. clock frequency $\leq$ 400 MHz
range:

ext. clock voltage   0.5 to 5 V_{pp}
swing:

The external clock input is ac-coupled—so the clock signal that you feed to this input must have a duty cycle of about 50%.

Setting the int/ext Period Ratio

For the calculation of the system clock period from the external clock period the following formula applies:

system clock period $= n \times$ external clock period,

where:

n =   1/8, 1/4, 1/2, 1, 2, 4, or 8 for 25 MHz pins and 100 MHz pins

n =   1/16, 1/8, 1/4, 1/2, 1, 2, or 4 for 50 MHz pins and 200 MHz pins.

n =   1/32, 1/16, 1/8, 1/4, 1/2, 1, or 2 for 400 MHz pins

Ratios other than these are not permissible.

The entry field int/ext period ratio gives the current value of n (see Figure 5-2). Use the fill menu to select the required ratio. The default setting is 1:1.

Example: If your external clock is running at a frequency of 360 MHz and you need a system clock frequency of 90 MHz, you would select 1:4 from the fill menu.
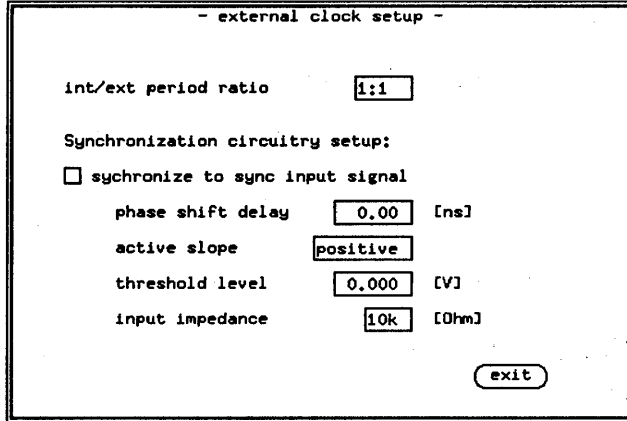
```
                    - external clock setup -


int/ext period ratio          [1:1  ]


Synchronization circuitry setup:

☐ sychronize to sync input signal

    phase shift delay      [  0.00 ]  [ns]

    active slope           [positive]

    threshold level        [ 0.000 ]  [V]

    input impedance          [10k ]  [Ohm]


                                    (  exit  )
```

Figure 5-2. External Clock Setup Dialog Box

Setting the Phase Shift Delay

To set up the required phase relationship, enter a positive decimal value into the phase shift delay entry field. Units are always nanoseconds. The system clock edge will be delayed by the entered value after the active slope of the Sync input signal has been detected. The default setting is 0.00 ns, the range is 0 to 999999 ns.

Setting the Sync Input Active Slope

The active slope of the Sync input signal can be positive or negative. Use the mouse to select the required slope from the fill menu. The default setting is positive slope.

Setting the Threshold Level

To set up the required threshold level of the Sync input, enter a decimal value into the threshold level entry field. The default setting is 0.000 V, the range is +10 V to −10 V.

Setting the Sync Input Impedance

The input impedance of the Sync input can be 10 kΩ or
50 Ω. Use the mouse to select the required impedance
from the fill menu. The default setting is 10 kΩ. If the
impedance is set to 50 Ω, the input threshold levels
should be reduced to the range ±5 Volt to prevent
damage to the tester hardware.

Selecting/Deselecting Phase Synchronization

Having set up the entry fields for the phase shift
delay, and Sync input active slope, threshold level
and input impedance, click the mouse on the check
box synchronize to external clock to toggle it. A
black check box means the system clock is synchronized.
A white check box means it is not synchronized. The
default is not synchronized.

Selecting Timing Definitions for Display

The menu bar option (Format) allows you to define the
format of the table of timing definitions. Through the
Define table format dialog box you can select timing
definitions of those device pins or groups of device pins,
that you wish to see in the lower part of the Timing
Setup Window (see Figure 5-1).

If you haven't selected any pins in the Define table
format dialog box, the table of timing definitions shows
the timing definitions of all pins that have been defined
in the Pin Configuration Window.

If you have selected some pins or pin groups in the
Define table format dialog box, then the timing
definitions of those pins or pin groups will be displayed
in the table of timing definitions.

Selecting All Configured Timing Definitions

The dialog box Define table format contains three browsers. The browser table format contains a list of all entries, be it pins or pin groups, currently selected for display in the table of timing definitions.

It also gives the type of entry:

i   input pin
o   output pin
g   group (of input or output pins)

As the browser can display only 15 entries at a time you have to use the scroll bar to see them all.

On the right of this browser there are the pushbuttons (set default) and (clear all), see Figure 5-3.
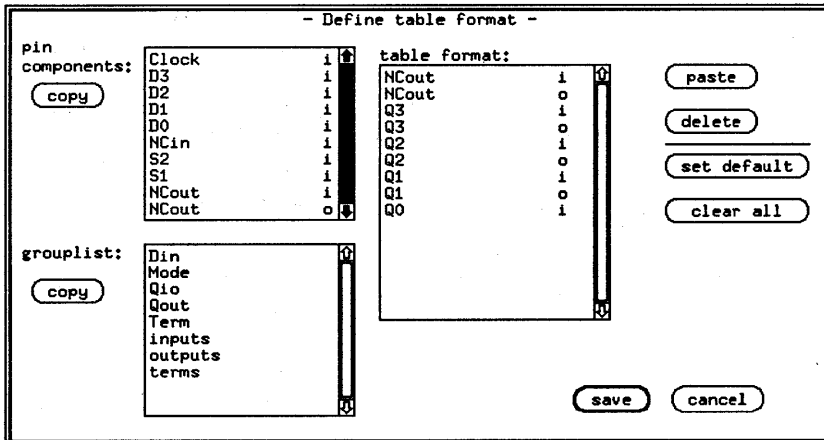
Figure 5-3. Define Table Format Dialog Box

To display the timing definitions of all configured timing pins, click the mouse on the (set default) pushbutton. The names of all configured pins also appear in the table format browser. Alternatively, you can click the mouse on the (clear all) pushbutton. This also displays the timing

definitions of all configured timing pins, but the `table format` browser will be empty.

These two conditions are the starting points for editing the entries in the table format browser. You can now delete some pins or pin groups from the table format browser or paste them to the browser.

Deleting Pins

To select a pin or a pin group for deletion from the table format browser, click it with the mouse. The selected entry will appear in inverse video. To select several adjacent entries, click the mouse on the first entry you wish to delete and without releasing the mouse button draw the mouse across all the entries to be deleted. The selected entries will appear in inverse video. To select several non-adjacent entries, hold down the (Shift) key and click the required entries with the mouse. The selected entries will appear in inverse video.

To delete the marked entries, click the (delete) pushbutton. The deleted entries are moved to the paste buffer. If you've made a mistake, you can get them back by clicking the (paste) pushbutton.

Adding Pins

Adding pins to the table format browser entails two steps:

1. copying pins or pin groups to the paste buffer (described in the next two sections)
2. pasting the buffer into the table format browser

Pasting into the Table Format Browser

The paste buffer contains pins or pin groups you have last copied into it. To append the contents of the paste buffer to the contents of the table format browser, just click the (paste) pushbutton.

To insert the contents of the paste buffer into the table format browser, select an entry in the browser where

you want the paste buffer copied. The selected entry will be displayed in inverse video. Now click the (paste) pushbutton. The paste buffer will be inserted before the selected entry.

Selecting Timing Definitions of Pins

The browser pin components contains all DUT pins that have been defined in the Pin Configuration Window and carry timing information.

To copy one or more pins to the paste buffer, select the pin(s) with the mouse and click the (copy) pushbutton. Use the mouse the same way as before.

Selecting Timing Definitions of Pin Groups

The browser grouplist contains all groups of DUT pins that have been defined in the Pin Configuration Window and carry timing information.

To copy a group or several groups into the paste buffer follow the same procedure as for pin components.

Example - Defining the Table Format:

Figure 5-4 shows the table format browser contains the input pins S2, S1 and the output pins Q3 to Q0. In order to add the pins D3 to D0 to the existing display in the table of timing definitions, such that they appear between pins S2, S1 and pins Q3 - Q0, you have to perform the following steps:

1. Mark the pins in the browser pin components, that you wish to display in the table of timing definitions - in this case the pins D3 to D0.
2. Mark the pin in the browser table format before which you wish to insert the extra pins - in this case Q3.
3. Click the mouse on the pushbutton (copy) next to the pin components browser to copy the marked pins into the paste buffer.

4. Click the mouse on the pushbutton ⌈paste⌋ to paste the buffer into the **table format** browser at the selected place.
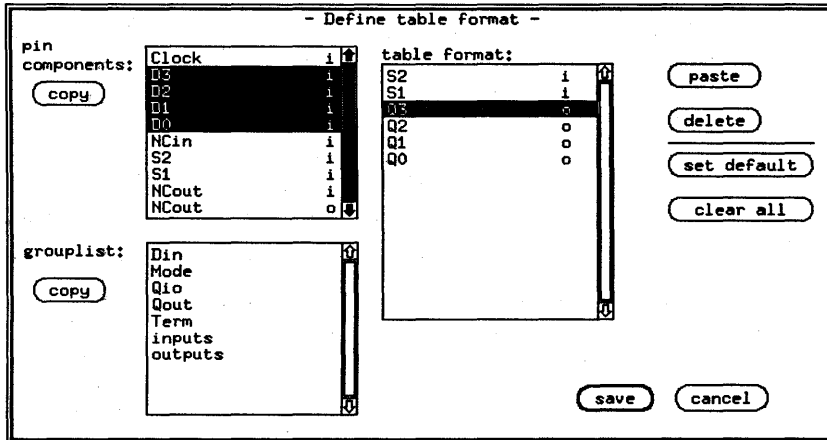
```
                         - Define table format -
pin
components:  ┌Clock            i█   table format:
            │ D3              i     ┌S2              i █
  ⌈ copy ⌋  │ D2              i     │S1              i            ⌈  paste  ⌋
            │ D1              i     │D3              o
            │ D0              i     │Q2              o            ⌈ delete  ⌋
            │ NCin            i     │Q1              o
            │ S2              i     │Q0              o            ─────────────
            │ S1              i                                  ⌈ set default ⌋
            │ NCout           i
            └ NCout           o█                                 ⌈ clear all ⌋

grouplist:  ┌Din              █
            │ Mode
  ⌈ copy ⌋  │ Qio
            │ Qout
            │ Term
            │ inputs
            │ outputs
            │                                            ⌈ save ⌋   ⌈ cancel ⌋
            █
```

Figure 5-4. Example - Defining the Table Format

Configuring a Pin Group

Any pin groups listed in the browser **grouplist** have been made up in the Pin Configuration Window. Once created there, they cannot be changed or edited in the **Define table format** dialog box. They can only be added to or deleted from the display of the table of timing definitions. If you wish to modify a pin group, you have to open the Pin Configuration Window.

The pin groups listed in the browser **grouplist** normally contain timing definitions of similar pins. For instance, all input data pins of a device could be grouped under the name D_{in}. If all these pins have the same timing characteristics, they can be combined into one group and the group then displayed in the table of timing definitions, rather than displaying the pins individually. This is the function of pin groups - purely to help you

edit the timing definitions. It is particularly useful when testing high pin-count devices.

Interactions between Pins and Pin Groups

Grouping similar pins together does not mean the individual pins in the group can no longer be displayed in the table of timing definitions. Both the individual pins and the pin groups made up of these pins can be displayed at the same time.

If you do this, you have to exercise a little care when modifying any of the timing parameters in the table of timing definitions.

Changing a Group Timing Definition . If you change any of the group timing definitions in the table (that is `format`, `leading edge` timing or `trailing edge` timing), all pin entries in that column relevant to that group will change to the new setting.

Changing a Pin Timing Definition. If you change any of the pin timing definitions in the table (that is `format`, `leading edge` timing or `trailing edge` timing), a question mark (`?`) appears in the changed column against the group containing the changed pin. This is because the group definition no longer agrees with the individual pin definitions in the group.

The hardware will still function in this condition, since the individual pin definitions determine the state of the hardware, not the group definitions.

If you need to have different timing definitions in a pin group, and you don't wish to have undefined pin groups in your Timing Setup, you'll have to open the Pin Configuration Window and remove the undefined group. Remember though, that a pin group includes not just the timing definitions for all its pins, but the level and vector definitions as well.

6
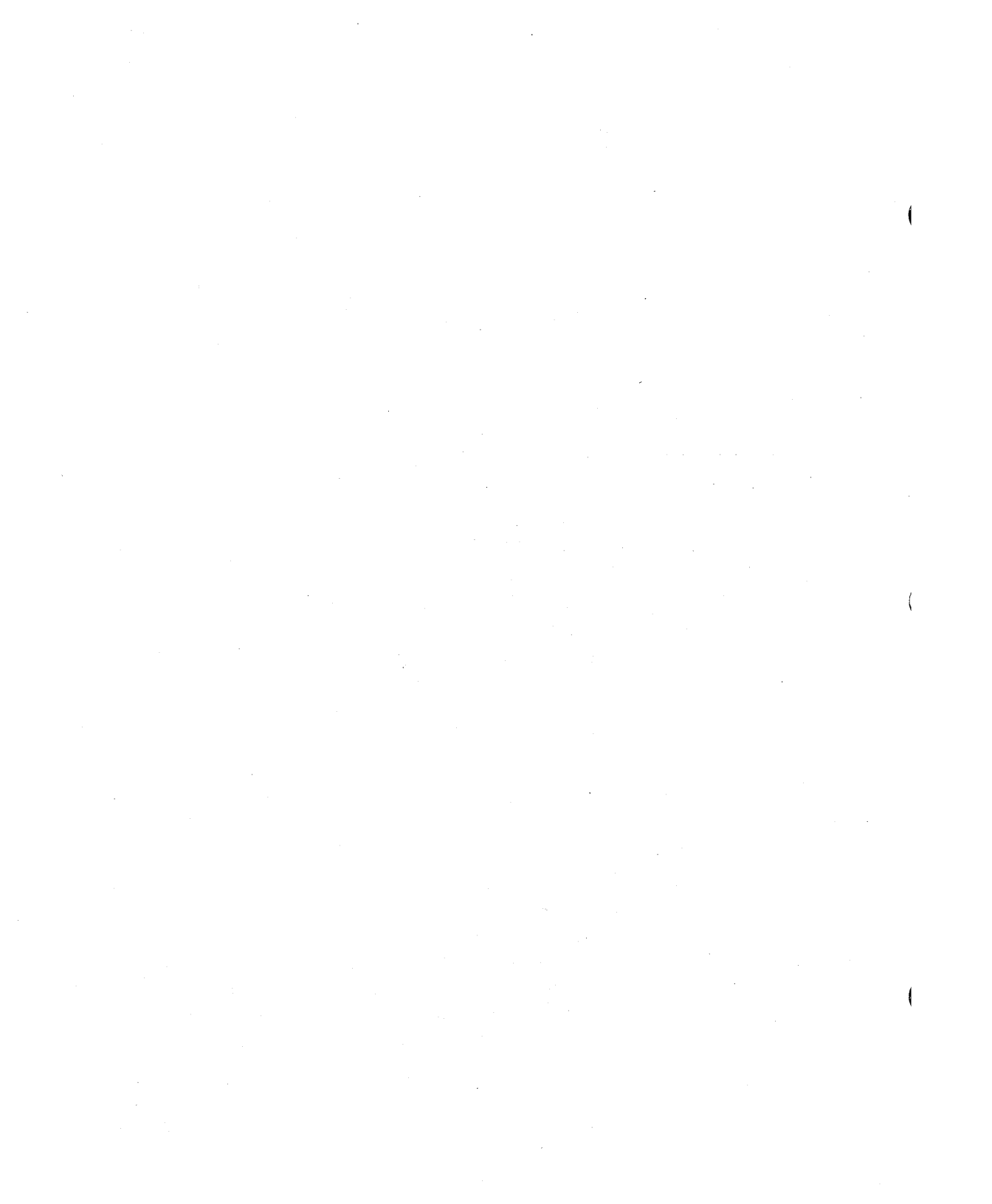
Editing Vectors

The vector editor is designed more for handling existing vector data than for generating new data. Any large tests that you run probably will be based on simulation data developed on a CAE system, and loaded into the system using CAE link software. You can, however, edit any of the imported vector data in the same format as it is displayed.

You can also add vectors to existing data, using the built-in Pattern Generator. This facility enables you to generate repetitive data bit sequences or vector patterns - either to overwrite existing data or to create additional vector addresses.

The Vector Sequencer provides a convenient interface for applying your test vectors to the DUT. You can specify a series of blocks to be transferred to the DUT, and conditional branching in the flow of test vectors allows you to develop quite complex test structures.

All the data necessary to run a functional test is stored in one file, and it is this file that is updated when you use the vector editor. The vector file is always stored in the **vectors** subdirectory of the device that you select. The vector file contains the following:

- vector sequencer instructions
- mask-, break-, and expected vectors
- vector data
- table format

The Vector Setup Window is opened by clicking the Vectors softkey on the system main menu. If you are using the demo device to acquaint yourself with the system, and if you have loaded one of the functional tests supplied, the Vector Setup Window will be displayed similar to the one shown in Figure 6-1.

Pin/Group Names	Clock	NCin	Mode	Din	Qout	NCout	Q3	Q2	Q1	Q0	D2	Comments
Pin Mask	A	A	A	AAAA	A	X	A	A	A	A	A	
	DD	DD	DD	DD	ED	ED	ED	ED	ED	ED	DD	
Exp. Vec BreakVec	0	H	H	HHHH	F	1	1	1	1	1	H	
12	1	0	2	0000	C	1	1	1	0	0	0	
13	1	0	2	0000	D	1	1	1	0	1	0	
14	1	0	2	0000	E	1	1	1	1	0	0	
15	1	0	2	0000	F	0	1	1	1	1	0	
16	1	0	1	0000	E	1	1	1	1	0	0	Start Downcount
17	1	0	1	0000	D	1	1	1	0	1	0	
18	1	0	1	0000	C	1	1	1	0	0	0	
19	1	0	1	0000	B	1	1	0	1	1	0	
20	1	0	1	0000	A	1	1	0	1	0	0	
21	1	0	1	0000	9	1	1	0	0	1	0	
22	1	0	1	0000	8	1	1	0	0	0	0	
23	1	0	0	0111	7	0	0	1	1	1	1	Reset Cycle
24	1	0	1	0000	6	1	0	1	1	0	0	
25	1	0	1	0000	5	1	0	1	0	1	0	
26	1	0	1	0000	4	1	0	1	0	0	0	
27	1	0	3	0000	4	1	0	1	0	0	0	
28	1	0	3	0000	4	1	0	1	0	0	0	
29	1	0	3	0000	4	1	0	1	0	0	0	
30	1	0	3	0000	4	1	0	1	0	0	0	
31	1	0	1	0000	3	1	0	0	1	1	0	

Figure 6-1. The Vector Setup Window

As you can see, the window is divided roughly into four areas:

- pin/group names
- special vectors
- drive and expected vectors
- comments field

How Pins and Groups are Displayed

The pin and group names are displayed along the top of the window with the pin mask displayed directly underneath them. All pins defined in the vector setup can be displayed. To the right of the vector data is an area for adding comments, for annotating each line of the vector table.

The number of pins to be displayed, and the width of the comment field, are set in the **Define table format** dialog box (refer to "Setting the Display Format"), and you can list any combination of lists and groups that you want.

The horizontal black bar at the bottom of the window shows where you are within this list, while the width of the bar shows how much of the list is currently displayed. Click the scroll bar to move through the list as you want.

The **Select** pull down menu allows you to view more of the *drive* and *expected* vectors at one time. You can increase the height of the Vector Setup window by selecting **Double_size**. If the double size window is selected, you can also choose the **show_full_names** option.

The three *special vectors* are displayed directly below the pin and group names:

- pin mask
- break vector
- expected vector

You can suppress the display of the three *special vectors* by clicking **Hide special vectors** option in the 'Select menu.

What the Pin Mask Does

The pin mask determines which pins will be ignored during a functional test. The pins that are active are marked with an A while pins that will be ignored are marked with a Z (input pins) or an X (output pins). Input pins are set to high impedance.

The next row shows whether the pins are drive data for input pins (DD) or expected data for output pins (ED).

What the Break Vector Does

The break vector allows you to set the state of the DUT input pins while the tester is in the break state. The tester is in the break state while the hardware is in the Connected state (shown in the top bar of the Report Window), and no test is presently running. During this time, the break vector that you specify is continually sent to the DUT.

Setting the Break Vector

You can set each bit of the break vector to have one of the following states:

- 0 - a logic 0 level;
- 1 - a logic 1 level
- H - to hold the value that was last output is held) in the Special Vectors part of the Vector Setup Editor. This is an advanced topic, which is discussed in detail in the manual *Advanced Testing with the HP 82000*.

Break Cycles

The break vector is also sent to the DUT during *break cycles*. Break Cycles are discussed in "Break Cycles."

Note

The DUT outputs are not read during the break cycles.

What the Expected Vector Does

In Real Time Compare (RTC) mode, the DUT data is continuously compared to the expected data (that you enter in the Vetor Setup screen) The results from these comparisons are ORed together to give a pass/fail indication for each vector. It is this pass/fail condition that causes test flow branching (conditional jumps) in the sequencer.

In Data Acquisition (DA) mode, there is no comparison with the expected data, rather, the DUT data is captured and then later transferred to the controller for analysis. To enable conditional branching in DA Mode, the data from the DUT, as well as being stored in the receive pattern memory, is also continuously compared to the Expected Vector. This generates an error condition for the sequencer in the same way as they are generated in RTC Mode.

If you are using this facility, it would be normal to specify a *jump on no fail* condition for the sequencer instruction, that is, the sequencer would perform a jump when the DUT outputs the expected vector. Further details about the operation of the Vector Sequencer are given in the manual *Advanced Testing with the HP 82000*.

One application of this would be testing a device whose outputs need to be in a known state before the test starts. Initialization vectors could be fed to the device until the DUT is in the expected state. The appearance of the Expected Vector would then trigger the start of the relevant test sequence.

How the Vector Data is Displayed

Vector data is listed in vector-order starting at address 0, and finishing with the highest address. The height of the vertical scroll bar shows what fraction of the entire vector address is being displayed. Moving the scroll bar allows you to scroll through the entire vector range. Likewise, moving the horizontal scroll bar allows you to scroll across the pin field.

Setting the Display Format

You may be wondering what order the pins and groups are displayed in. You select the pins you want to display, their order and their format yourself. Click **Format** on the menu bar, and select the **def.tab_format** option. A dialog box appears similar to that shown in Figure 6-2.
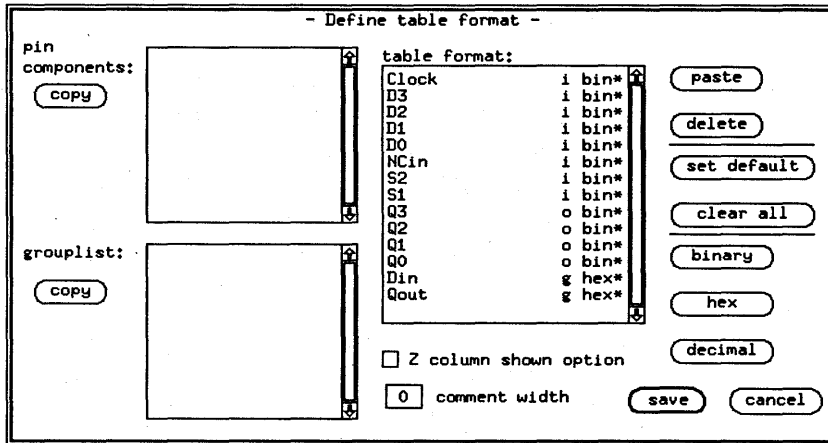
Figure 6-2. Defining the Format for Displayed Vectors.

What You Can and Cannot Edit

You should first understand the conditions under which you can edit pins and groups. In order to make the system as flexible as possible, the Pin Configuration Window places few restrictions on the device pins that you can group together.

However, some combinations of pin types within a group can cause inconsistencies when you are editing vectors. So to avoid accidentally introducing configuration related bugs (which would be difficult to trace), the vector editor prevents you from editing groups which contain:

■ a mixture of I and O pins;

■ DPS pins;

■ a mixture of operating modes;

■ active termination components.

The chapter Configuring Pins for Testing describes these restrictions in detail.

Due to the nature of the display, the following restrictions exist for the size of groups that can be edited:

■ **binary format:** not more than 63 pins.

■ **hex format:** not more than 252 pins.

■ **decimal format:** not more than 32 pins.

Note

Groups larger than those listed above cannot be edited in the vector setup window. They can still exist and their individual components (pins) can be edited.

The groups or pins that cannot be edited are displayed in gray scale in the pin components and grouplist browsers.

If a group is eligible for editing and you later change a pin's configuration such that it is no longer eligible, it will remain in the table format browser, but will be displayed with an asterisk next to its name. Attempts to edit the group will be unsuccessful.

Pasting Pins and Groups into the Window

When looking at the vector data for a test, it will not normally be possible to display all test pins in one window. The `define table format` window is a flexible utility that lets you display pins or groups that you want, in the format that you want them.

Figure 6-2 above shows the layout of this window. The window has three browsers labeled `pin components`, `grouplist`, and `table format` respectively. The two browsers on the left show you all the configured pins and groups for this device. The right browser contains what will be displayed in the Vector Setup Window.

You can paste any number of pins into the table format browser, in fact, if you click the `set default` pushbutton, all configured pins will be pasted into this browser. Because they will be copied as single pins, they will all be allocated a binary format.

Similarly, if you want to define a new format, clicking the `clear all` pushbutton will clear the `table format` browser completely. Remember that nothing that you do in this window can affect your configuration or data; it simply lets you display the vector data as you want it.

There might be occasions when you want to display a pin or group more than once. For example, you might be interested in one particular pin of an address bus. In this case, you could display the bus as a group with hex format, and at the same time, display one pin alongside it in binary format.

Cutting and Pasting

As with the other windows, when you want to cut and paste, you need to do the following.

1. **Mark where you want the items copied.** If you do not mark a position in the `table format` browser, the contents of the paste buffer will be copied at the end. If you mark a position in the browser, the contents of

the paste buffer will be inserted before that entry. Mark an entry in the table format browser by clicking it with the mouse.

2. **Define what it is that you want to copy.** Whether you are copying groups or pins, you define them as follows. First click the pins or groups that you want to copy. If the pins are listed next to each other, click the first one and without releasing the button, drag the mouse to the last one. If the items do not lie next to each other, select them one after the other by holding down the (Shift) key and pressing the mouse button. The selected items will be displayed in inverse video. Note that you can only copy from one browser at a time. Click the copy pushbutton for whichever browser you are using. This copies your selection into the `table format` browser paste buffer.

3. You can repeat the first copy process by clicking the `paste` pushbutton and the contents of the paste buffer will be written into the `table format` browser. Note that the paste buffer retains its contents until it is updated with another copy. Each time that you click the `paste` pushbutton, the buffer contents are written into the browser at the marked position.

Look at the `table format` browser shown in Figure 6-2 above. The pin or group name for each entry is displayed on the left of the browser. The format (`bin`, `hex`, or `dec`) is displayed on the right, and shows whether the format is binary, hexadecimal, or decimal. A single letter in front of the format tells you which type of pin is listed. This is either `i`, `o`, or `g`, and shows whether the entry is an input pin, an output pin, or a group of pins.

Changing the Display Format

To change the format of pins or groups, simply select the ones you want you change in the `table format` browser,

and click the appropriate pushbutton; binary, hex, or decimal. Remember that you can change the format of blocks of entries in the browser by simply marking the block and clicking the appropriate field.

Displaying Tristate Vectors

In the Vector Setup window, the tristate setting for each pin or group is displayed to the right. The tristate column itself is headed with a Z. If you want to display tristate information, click the Z format pushbutton. When the field is black, tristate information will be displayed.

Displaying the Comment Field

You can annotate each line in the vector table with a comment of up to 60 characters long (including spaces). These comments are displayed at the right hand edge of the Vector Setup window, and you can set the width of this field using the comment width box in the Define table format window.

The maximum field width is 60 characters. You can suppress the comments from being displayed, by entering a width value of 0 in the box. When you pull down the Vector Setup window, and no vectors have yet been set, the comment field defaults to a width of 10 characters.

Saving Format Files

When you are performing a variety of tests on a device, you may find it useful to have several different table format definitions readily available.

To save your table format definition for later use, click the File option on the menu bar, and in the pull down menu select either save or save as. Figure 6-3 shows the dialog box.

```
┌─────────────────────────────────────────────┐
│  ○ save all                                   │
│  ● save partial                               │
│                                               │
│     ■ sequencer data                          │
│     ■ mask-, break-, expected vector          │
│     ■ vector data                             │
│     ■ table format              ┌──────────┐  │
│  save file                      │   save   │  │
│                                 └──────────┘  │
│     ┌─────────────────────┐     ┌──────────┐  │
│     └─────────────────────┘     │  cancel  │  │
│                                 └──────────┘  │
└─────────────────────────────────────────────┘
```
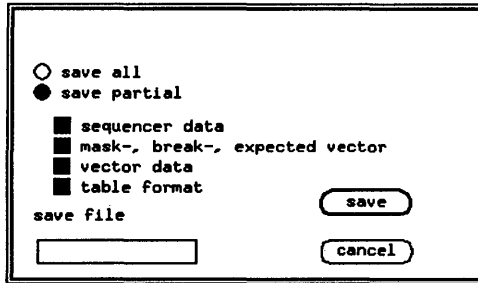
Figure 6-3. Save/Save as Dialog Box

Select the options to be saved, and when your selection
is complete, click the **save** pushbutton.

Editing the Vector Data

The vector data can be freely edited as described in
Chapter 1. The values that you enter depend on the
input signal formats, and the mode of comparison for the
output signals.

Drive Data Values

The values that you can enter for drive data are

- 1 for a logic high
- or 0 for a logic low.

Tristate Values

The tristate bit determines whether the pin should be
driven to tristate during the clock period, or whether it
should be active. The Tristate condition is controlled by
the trailing edge entry field of the Timing Setup window.

z   the pin will be driven to tristate at the
            next trailing edge. If the pin is already
            in tristate, it will be held there.

| 0 | the pin will be driven to the value specified by the drive data bit during this period. If the pin is already active, this tristate value will not affect it, it will remain active. |

In 25 and 100 Modes, the tristate column is displayed for all pin formats, and you can edit this data. For R1, RC, and RZ formats, this will have no effect on the test because for these formats, the tristate portion of the drive pattern memory is not used.

In the other modes, the tristate column is not displayed for any pin format.

Expected Data Values

You can define four possible states for an expected vector:

0	the DUT output must be lower than the logic low level (specified in the Level Setup Window) for the comparison to pass.
1	the DUT output must be higher than the logic high level (specified in the Level Setup Window) for the comparison to pass.
X	Don't care. No comparison of the signal level will be made in this period.
i	Intermediate. The level should be between the high and low levels specified.

If the test mode is changed to 50, 200 or 400, the only valid values that you can specify are 1 or 0. In this mode, only one bit of the receive pattern memory is allocated for each period.

You can however specify the full range of expected values if a channel multiplexing mode or 200COM mode is selected.

Setting Blocks of Vectors

The Edit pulldown menu provides you with three dialog boxes for setting vector data for individual pins or groups of pins within a range of vector addresses. The dialog boxes are:

- set DD_bits for setting data bits to drive the DUT inputs;
- set ED_bits for setting the data expected from the DUT outputs.
- and generate_pattern for setting more complex patterns of bits over the selected range.

Setting Drive Data Bits

Drive data bits are set using the set DD_bits dialog box, which is shown in Figure 6-4. This dialog box allows you to set a block of drive vectors for a selection of pins or groups.
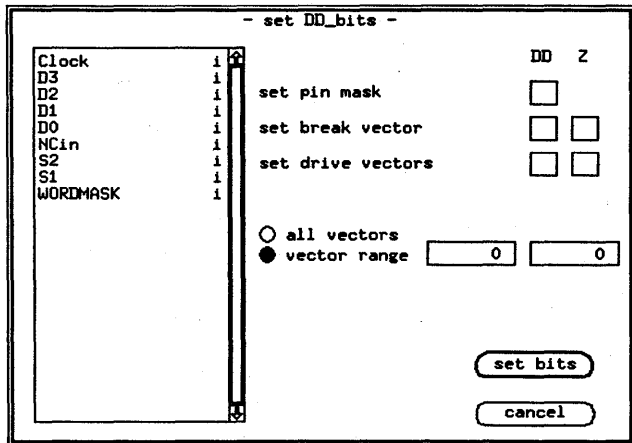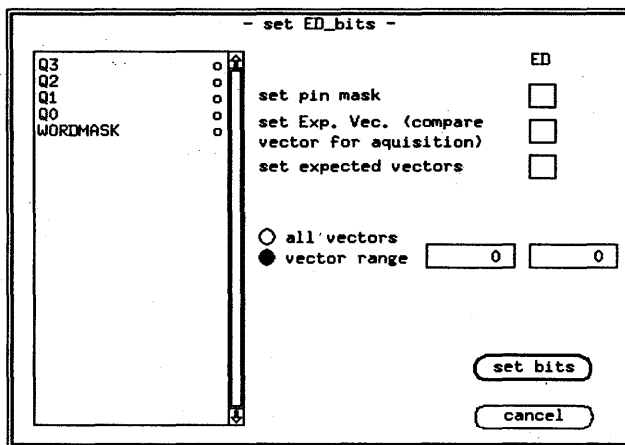
Figure 6-4. Setting Drive Data Bits

The browser on the left of the box shows all the
pins/groups that have been defined in the table format
dialog box. In this case the device is the MC10136
demonstration device.

You select the pins or groups to be set with the mouse,
and then make the selections you want for the remaining
fields in the dialog box.

set pin mask

The pin mask, as previously mentioned, is used to mask
selected pins or groups from testing. When a drive pin is
masked, the pin itself is set to tristate for the duration of
the test.

The values that can be set are A (active) or Z(tristate).

set break vector

The break vector is continually output to the DUT while
the sequencer is in the break state. The break vector is
set in the same way as the drive vectors, with an extra

state H. H stands for *hold* which holds the pin at its last value defined before the system entered the break state.

set drive vectors

There are two edit fields allocated for setting drive vectors (and the break vector). The left field sets the drive data for the block and the right field sets the tristate data.

The values that you can enter for drive data are always 1 or 0. The way this data is interpreted depends on the signal format. With the RZ, RC, and R1 formats, each bit represents an input pulse.

The tristate bit determines whether the pin should be driven to tristate during the clock period, or whether it should be active.

Z   the pin will be driven to tristate at the next trailing edge. If the pin is already in tristate, it will be held there.

0   the pin will be driven to the value specified by the drive data bit during this period. If the pin is already active, this tristate value will not affect it. It will remain active.

vector range

Having set the vector data, you need to set the range of vectors. If you want to set all vectors to these values, click the all vectors button.

If you wish to set a range of vectors, fill in appropriate values in the vector range entry fields.

Once you are satisfied that the parameters are correct, click the set bits pushbutton to accept the changes.

Note 👆

The changes in this dialog box only affect the vectors
currently in the controller memory. No changes to the
original vector file will be made until it is overwritten
with the **save** option.

To ignore the changes you just made, press the (cancel)
pushbutton.

Setting Expected Data Bits

Expected data bits are set using the set ED_bits
dialog box, which is shown in Figure 6-5. This dialog
box allows you to set a block of expected vectors for a
selection of pins or groups.

```
┌────────────────────────────────────────────────────┐
│                   - set ED_bits -                    │
│  ┌──────────────┐ o ▲                           ED   │
│  │Q3      ...   │ o                                   │
│  │Q2            │ o   set pin mask              ☐     │
│  │Q1            │ o                                   │
│  │Q0            │ o   set Exp. Vec. (compare    ☐     │
│  │WORDMASK      │ o   vector for aquisition)          │
│  │              │     set expected vectors      ☐    │
│  │              │                                     │
│  │              │     ○ all'vectors                   │
│  │              │     ● vector range  [   0  ] [   0 ]│
│  │              │                                     │
│  │              │                                     │
│  │              │                     (  set bits  )  │
│  │              │ ▼                                   │
│  └──────────────┘     (   cancel   )                 │
└────────────────────────────────────────────────────┘
```

Figure 6-5. Setting Expected Data Bits

As you can see, the dialog box is similar to the set DD
bits box. The browser again contains the pins that have
been defined in the **define table format** dialog box.
After selecting the pins or groups to be set, you select
their values in the same way as for the drive data.

set pin mask

The pin mask, as previously mentioned, is used to mask selected pins or groups from testing. When an output pin is masked, its data does not participate in the compare result during testing.

The values that can be set are A (active) or X (don't care - masked).

set Exp. Vec.

The expected vector is compared continuously with the data received from the DUT in the Data Acquisition Mode (refer to "What the Expected Vector Does"). An expected vector bit can have one of four states. These states are selected with the valid field entries:

0   the level detected should be lower than the level specified in the Level Setup Window.

1   the level detected should be higher than the level specified.

X   Don't care. No comparison of the signal level will be made in this period.

i   the level should be between the high and low levels specified.

set expected vectors

The permissible values for expected vectors depend on the mode of testing. If a pin operates at 50, 200, or 400 mode then only 0 or 1 can be set.

If the mode of testing is 25, 50MUX, 100, 200COM, 200MUX or 400MUX, the values 0, 1, X, or i. can be set. These values are described above in the section on "What the Expected Vector Does."

vector range

You select the vector range in the same way as for the drive data (refer to "vector range" above).

Once you are satisfied that the parameters are correct, click the **set bits** pushbutton to accept the changes.

Note 👆 The changes in this dialog box only affect the vectors currently in the controller memory. No changes to the original vector file will be made until it is overwritten with the **save** option.

To ignore any changes you may have made, press the (cancel) pushbutton and the previous state will be restored.

Generating Vector Bit Patterns

The Pattern Generator facility enables you to generate a pattern of bits over a selected range of vector addresses for selected pins. The Pattern Generator is selected by clicking **generate_pattern** in the Vector Setup **Edit** menu. Figure 6-6 shows the dialog box used to program the Pattern Generator.

```
┌─────────────────────────────────────────────────────────────┐
│               - generate vector pattern -                     │
│ ○ insert                                                      │
│ ● overwrite              ● set to one  ○ set to zero          │
│                          ○ up counter  ○ down counter         │
│ ┌──────────────┐─┬─┐     ○ walking one ○ walking zero         │
│ │Clock      i│▓│     ○ repeat      ○ checker board       │
│ │NCin       i│ │     ○ complement  ○ random              │
│ │Mode       g│ │                                         │
│ │Din        g│ │                                         │
│ │Qout       g│ │                                         │
│ │NCout      o│ │                                         │
│ │Q3         o│ │                                         │
│ │Q2         o│ │                                         │
│ │Q1         o│ │                                         │
│ │Q0         o│ │     ○ all lines                         │
│ │D2         i│ │     ● line range    ┌──────┐ ┌──────┐   │
│ │           │ │                      │   0  │ │   0  │   │
│ │           │ │     ○ all pins                           │
│ │           │ │     ● selected pins  ( gen_patt )        │
│ │           │ │                                          │
│ │           │▓│                      ( cancel )          │
│ └──────────────┘─┴─┘                                          │
└─────────────────────────────────────────────────────────────┘
```
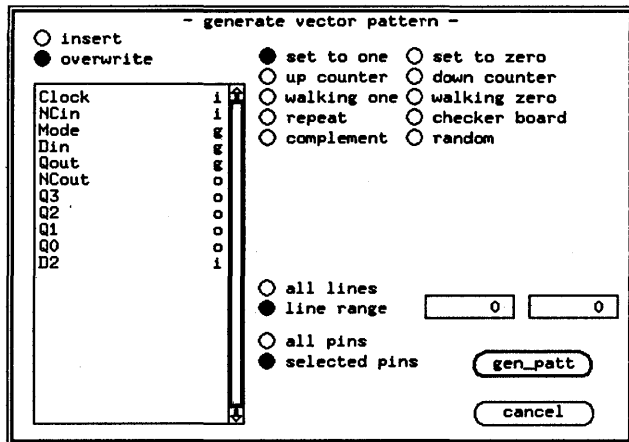
Figure 6-6. Using the Pattern Generator

The dialog box operates in a similar way to the boxes for
set DD_bits and set ED_bits. The browser shows the
pins and pin groups defined in the define table format
dialog box.

Select the range of vector addresses for the pattern using
the line range entry fields or by selecting the all
lines radio button.

Indicate the pins to be modified by using the mouse to
select individual pins from the browser, or by clicking
the all pins radio button. If you make a mistake while
choosing pins, click the selected pins radio button to
deselect all the pins in the browser.

The overwrite option replaces the existing data in the
selected address range with the generated pattern.

The insert option places the pattern into the selected
address range but shifts the data already in these
addresses (and all the following data) up the address
range. With this option, no data is lost.

The Pattern Generator facility provides ten different pattern options. Select the pattern you want by clicking one of the pattern radio buttons. When you are satisfied that the pattern and the vector ranges are correctly defined, actuate the pattern generator by clicking the (gen_patt) pushbutton.

To abort your pattern selection at any stage, click the (cancel) pushbutton.

set to one

The set to one option sets all the selected vectors to 1, irrespective of the previous values in the selected range.

set to zero

The set to zero option sets all the selected vectors to 0, irrespective of the previous values.

up counter

The up counter option provides an additional entry field labeled relative displacement. Enter the number at which you want the count to start.

The pins chosen in the browser are read as the individual bits in a binary word - with the left-most pin as the most significant bit and the right-most pin as the least significant bit.

For the first address in the selected vector range, the up counter option enters the binary value of relative displacement. For each successive address, this value is incremented by one, until the last vector address in the range is reached.

down counter

The down counter produces a pattern which is the *complement* of the up counter pattern: the first vector address is given a value

$$2^{\text{no. of pins}} - (\text{ relative displacement} + 1)$$

This value is then *decremented* through the addresses in the selected range.

walking one

The walking one option enters a 1 in the starting address vector of one of the selected pins, and 0 for all the other pins. In the following vector address, a 1 is only entered under the next selected pin to the right.

This sequence continues successively through the addresses until the right-most selected pin has been made 1. The 1 then walks back the other way - as each pin to the *left* is, in turn, made 1 until the left-most pin is reached. The entire pattern is repeated until the last address in the vector range is reached.

The entry field initial column allows you to specify from which pin the walking one will start (the left-most pin selected will be column 0).

Figure 6-7 shows an example dialog box setup. The vector range for the pattern starts at address 10 and finishes at address 20. The selected pins are (from left to right) Q3, Q2, Q1, and Q0. Column 2 (pin Q1) will be the first pin in the pattern.

```
                    - Pattern Generation -
O insert vectors
● overwrite vectors    O set to one    O set to zero
                       O up counter    O down counter
┌──────────────┐       ● walking one   O walking zero
│Clock      i ▲│       O repeat        O checker board
│D3         i  │       O complement    O random
│D2         i  │
│D1         i  │
│D0         i  │
│NCin       i  │
│S2         i  │
│S1         i  │           initial column        ┌──────┐ 2
│Q3         o  │                                  └──────┘
│Q2         o  │
│Q1         o  │       O all vectors
│Q0         o  │       ● vector range  ┌────┐ 10  ┌────┐ 20
│WORDMASK   o  │                       └────┘     └────┘
│           ▼ │
└──────────────┘       O all pins
                       ● selected pins    ( PG )

                                          ( cancel )
```
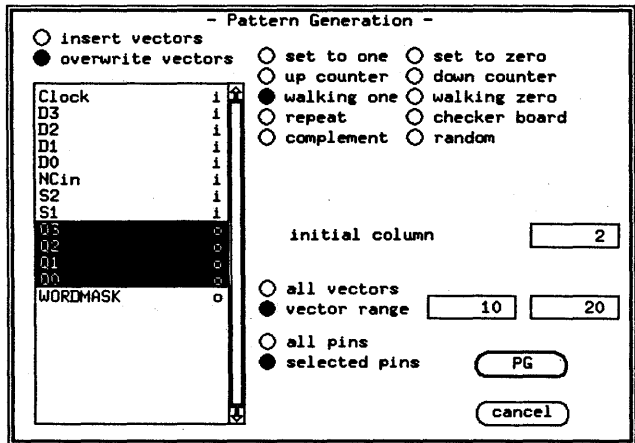
Figure 6-7. Defining a Walking One Vector Pattern

Figure 6-8 shows the pattern defined by this dialog box.

File Edit Select Format

Pin/ Group Names	Q 3	Q 2	Q 1	Q 0	Q o u t
	ED	ED	ED	ED	ED
9	0	0	0	0	0
	0	0	0	0	0
10	0	0	1	0	2
	0	0	0	1	1
11	0	0	1	0	2
	0	1	0	0	4
12	1	0	0	0	8
	0	1	0	0	4
13	0	0	1	0	2
	0	0	0	1	1
14	0	0	1	0	2
	0	1	0	0	4
15	1	0	0	0	8
	0	1	0	0	4
16	0	0	1	0	2
	0	0	0	1	1
17	0	0	1	0	2
	0	1	0	0	4
18	1	0	0	0	8
	0	1	0	0	4
19	0	0	1	0	2
	0	0	0	1	1
20	0	0	1	0	2
	0	1	0	0	4

Figure 6-8. An Example Walking One Pattern

walking zero

The walking zero option produces a similar pattern to walking one, but with a 0 in one column of each vector address. The initial column entry field functions in the same way as with the walking one option.

repeat pattern

The repeat pattern option allows you to repeatedly copy a sequence of vector values into the selected line range.

Enter the start and finish addresses of the bit sequence to be repeated in the Pattern: line range entry fields.

checker board

The checker board option alternately enters 1 and 0 across the pins of each address in the vector range. For each selected pin, the entered value also alternates 1 and 0 through the selected address range, thus creating a 'checker-board' pattern of 1s and 0s.

If you enter a zero or an even number in the entry field initial field, a 1 appears in the top left corner of the checker board (i.e. in the first selected address of the first selected pin). If an odd number is entered as the initial field, the top left corner is a 0.

Figure 6-9 show an example checker board pattern.

Pin/ Group Names	Q 3	Q 2	Q 1	Q 0	Q o u t
	ED	ED	ED	ED	ED
0	0	1	0	1	05
	1	0	1	0	10
1	0	1	0	1	05
	1	0	1	0	10
2	0	1.	0	1	05
	1	0	1	0	10
3	0	1	0	1	05
	1	0	1	0	10
4	0	1	0	1	05
	1	0	1	0	10
5	0	1	0	1	05
	1	0	1	0	10
6	0	1	0	1	05
	1	0	1	0	10
7	0	1	0	1	05
	1	0	1	0	10
8	0	1	0	1	05
	1	0	1	0	10
9	0	1	0	1	05
	1	0	1	0	10
10	0	1	0	1	05
	1	0	1	0	10
11	0	1	0	1	05
	1	0	1	0	10

Figure 6-9. An Example Checker Board Pattern

complement

The `complement` option swaps all the 1s in the selected vector addresses for 0s and vice-versa.

random (PRBS)

The `random (PRBS)` option invokes the Pseudo Random Binary Selector (PRBS) - a complicated algorithm which generates data with no apparent logical order. This 'random' data pattern is then entered into the selected address range for the selected pins.

In the `PRBS start` field you can enter any non-zero number with up to seven digits. This number triggers the PRBS algorithm.

Copying Vectors

The copy vectors dialog box, accessed from the Edit pull-down menu, allows you to copy vector data or comments from one block of consecutive vector addresse (the *source*) to another identically sized block (the *destination*). The dialog box is displayed similar to Figure 6-10.
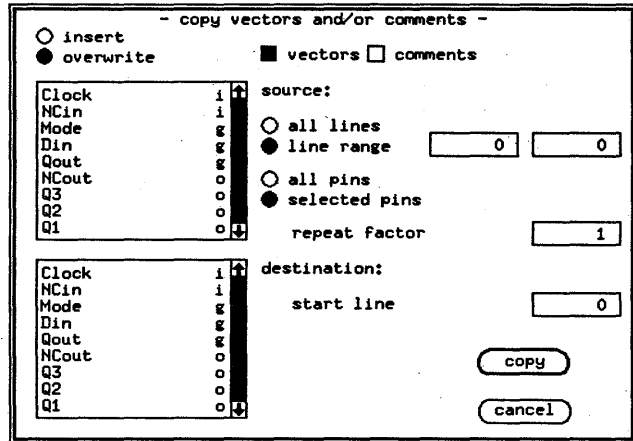
Figure 6-10. Copying Vector Data

Selecting the Line Range

The dialog box contains two browsers, marked source and destination, both containing all the currently defined pins and groups. Select the source and destination pins by clicking them in these browsers.

Enter the start and finish address of the source block in the line range entry field. Alternatively, if you click the all vectors radio button, every vector address for the selected pins will be copied.

Repeat Factor

The value you enter in the repeat factor entry field determines how many times the source block is copied to

the destination. If a factor greater than one is selected, the copied blocks of data are entered into consecutive blocks of addresses beginning with the `destination: start vector.`

Insert and Overwrite mode

Two radio buttons at the top of the dialog box allow you to toggle between `insert` mode and `overwrite` mode.

The `insert` mode copies the source vector block into a corresponding sized block of addresses beginning with the destination `start line`. The vector data already in these lines and in the all following lines is shifted up the address range. No vectors are lost.

The `overwrite vectors` mode copies the source block into a destination block of addresses beginning with the `start vector` *replacing* the existing data in these addresses.

Click the (copy) pushbutton to accept the changes, or the (cancel) button to ignore them.

Moving Vectors

The **move** option in the vector **edit** menu allows you to transfer a block of vectors for a selected group of pins from one address range to another. Moving a block of vectors has the same effect as inserting the group at the destination **start vector** while simultaneously deleting the block from the source addresses. Figure 6-11 shows the **move vectors** dialog box.
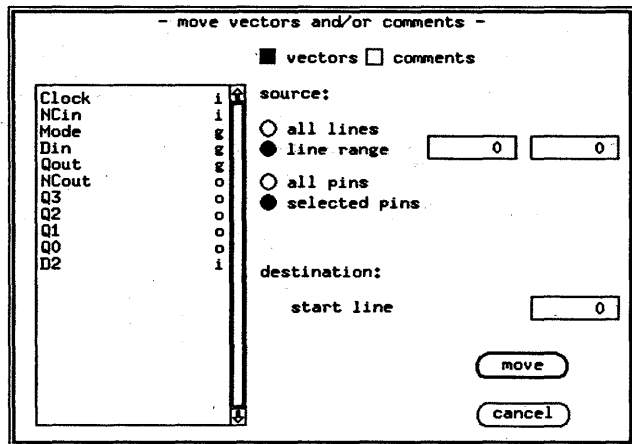
Figure 6-11. Moving Vector Data

Enter the start and finish address of the block be moved in the **line range** entry boxes (or click the **all vectors** radio button). From the browser, select the pins to be included in the block.

Then you must define the destination of the block to be moved. Enter the new starting address for the block in the **start vector** entry box.

Click the ⌾ move ⌿ pushbutton to accept the changes, or the ⌾ cancel ⌿ button to ignore them.

Searching and Replacing

The `search_and_replace` option, in the `Edit` pulldown menu, allows you to search through the vector table for:

- vector data patterns;
- messages in the `Comment` field;
- regular expressions in the `Comment` field.

You can also specify a replacement data string to substitute for the vector or comment strings that are found by the search.

Figure 6-12 shows the `search_and_replace` window.

```
┌──────────────────────────────────────────────────────────┐
│        - search&replace with vectors and/or comments -     │
│                                                            │
│   Search :            ■ vectors □ comments                 │
│                                                            │
│     with pattern          line range    [    0  ] [   0  ] │
│    Clock        i                                          │
│    NCin         i       ○ all pins                         │
│    Mode         g       ● selected pins                    │
│    Din          g                                          │
│    Qout         g                                          │
│    NCout        o       ○ all lines                        │
│                         ● line range    [    0  ] [   0  ] │
│                                                            │
│   Replace:            □ vectors □ comments                 │
│                                                            │
│     with pattern          line range    [    0  ] [   0  ] │
│    Clock        i                                          │
│    NCin         i       ○ all pins                         │
│    Mode         g       ● selected pins    ( srch&repl )   │
│    Din          g                                          │
│    Qout         g                                          │
│    NCout        o                          (  cancel   )   │
└──────────────────────────────────────────────────────────┘
```
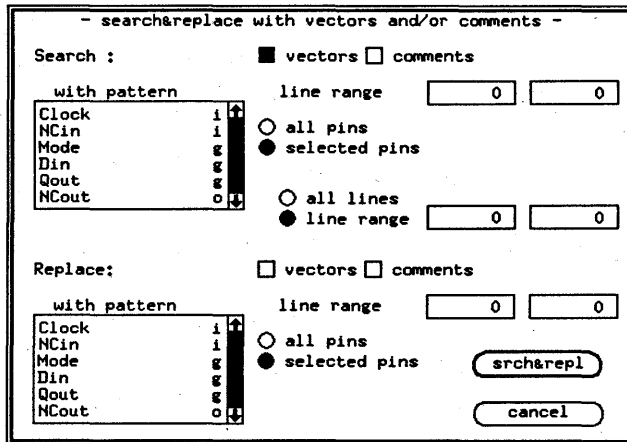
Figure 6-12. Search and Replace Window

Searching for Vectors or Comment Strings

To perform a search *without* replacing anything, you use the top half of the `search&replace` window to define your search string, while the `Replace:` fields (in the lower half) are left blank.

Each time that a match with your string is found, the line number is displayed in the `Report` window.

Performing a Search

1. Specify the type of data to be searched, by selecting either `vectors` and/or `comments` in the boxes at the top of the window.

2. Use the `Search: with pattern` browser to select which pins are included in the search.

3. For the pins you have selected, enter the `line range` in which the search pattern can be found.

4. Either: enter the beginning and end line numbers for the range of lines to search; OR click the `all lines` pushbutton.

5. Ensure that in the `Replace:` half of the screen, the boxes `vectors` and `comments` are both unfilled.

6. Click the pushbutton (srch&repl) to start the search.

As each match with your search pattern is found, the `Report` window prints the message:

`Pattern match: line` *line_number*

Searching Techniques

Searching using a String in an Unknown Location. If you do not already know the line numbers where you can find one occurrence of your search pattern, you can perform your search by creating a set of *dummy* vectors containing the pattern that you want to find. The procedure is as follows:

1. Move the cursor to the first free vector address (i.e. the bottom of the vector table).
2. Manually type in the pattern that you want to find, in the pin column that you wish to search.
3. Select `search_and_replace` in the `Edit` menu.
4. Perform the search (as outlined previously), using the line numbers that you have just created, as you search string.

When the search is complete, delete the lines that you have just created at bottom of the table.

Searching for Comments Using a Regular Expression. If you are searching for a comment, you can use the HP-UX notation for finding regular expression patterns. In this way, you can perform selective searches (e.g. using wildcards).

Full details of the pattern matching notation, and the metacharacters used, can be found in the *HP-UX Reference Manual.*

To use a regular expression for your search:

1. Enter the expression in the Comment field of the vector table, in a line which *does not* already have a comment.
2. perform a search, entering this line in the line range boxes, as your search string.

All lines which provide a logical match with the regular expression, are listed in the Result window.

When the search is complete, delete the regular expression from the Comment field.

Searching and Replacing Vectors or Comment Strings

This is done in a similar way to the search, but you must also select

- the pins to be used for the replacement,
- a range of lines from which the replacement data can be taken.

At each line where a successful match is found, the vector data at that location is overwritten by the replacement data that you have selected.

Performing a Search and Replace

1. Enter the search details in the top half of the window, as previously described in the section *Performing a Search.*

2. In the bottom half of the window, select either vectors and/or comments to be replaced.

3. From the Replace: with pattern browser, select the pin or pins to provide the replacement pattern.

4. Enter the line range. This is the range of lines which contain the data to be used as the replacement values.

5. Click the (srch&repl) pushbutton.

Setting the First Free Vector Address

The first free vector address ffvad defines the next free location in the pattern memory. This is normally set automatically to the next free address but it can be overridden. When overriding this setting, you alter the range of vectors that will be displayed. This does not have an effect on the pattern memory contents, only on the last vector address.

Note

When you save the vectors to a file, only those vectors between 0 and the *ffvad* are stored.

When you click the set ffvad option on the Edit pulldown menu, the set first free vector address box is displayed as shown in Figure 6-13.
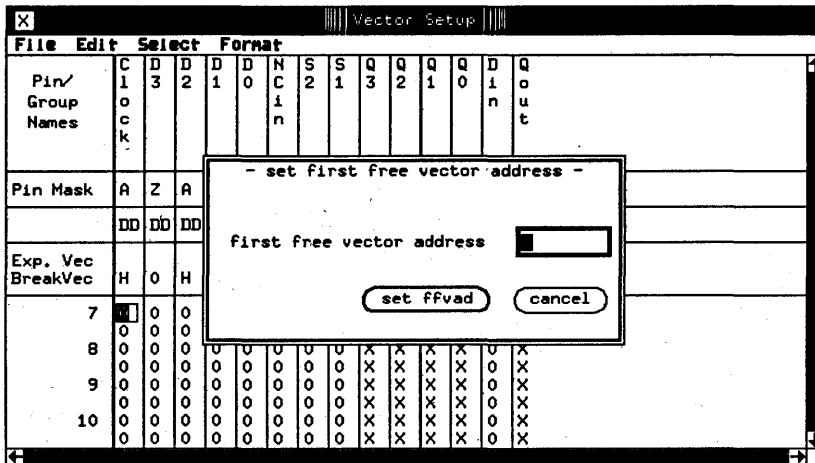
Figure 6-13. Setting the First Free Vector Address

Any value between 0 and the maximum vector depth can be set in the `ffvad:` edit box. Clicking the `set ffvad` pushbutton sets the new value for the first available address in the vector memory.

Using the Vector Sequencer Control Window

The vector sequencer allows you to perform tests using blocks of consecutive vectors. You can link vector blocks together in any order and specify conditional jumps and loops. The sequencer programs are interpreted by the system firmware, to control the addressing and transfer of test vector data.

Sequencer programming can be very complicated, and for this reason, two sequencer windows are provided; the Sequencer Control window and the Sequencer Programming window. You can access these two windows from the `select` pulldown menu in the Vector Setup window.

Both windows are translated into instructions, which specify;

- the start and finish address of a block of vectors,
- how they are to be sent to the DUT,
- and what action to take if either a fail or no-fail occurs when the expected and actual data is compared.

To start and stop the sequencer, you give a start label to your program and then specify this label in the Sequencer Start Label field of the test functions in the Test Control window.

The manual *Advanced Testing with the HP 82000* gives details about how to write complex test patterns using blocks of vectors, by using the Sequencer Programming window.

Break Cycles

Because of the nature of the digital shift registers used in the HP 82000, the output drivers generate the vector data *four* vectors at a time. If the number of vectors in your block is not a multiple of four, then *break vectors* are generated, in order to make this number up to a multiple of four.

During these *break cycles,* the break vector is sent to the DUT inputs. When this occurs, the result windows indicate BREAK instead of the vector number.

If a failure occurs during the *break vectors* (shaded areas in Figure 6-15, Figure 6-16 and Figure 6-17) it is not detected by the sequencer until the next test has begun. The sequencer then enters the break state at the end of that test.
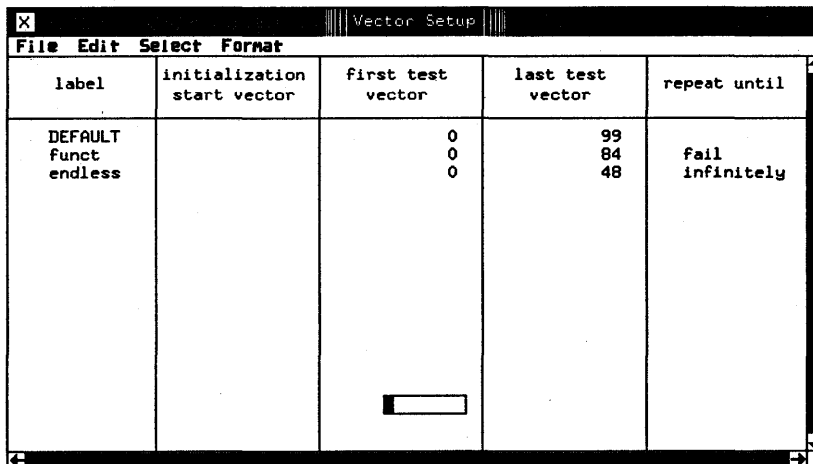
The occurrence of Break Cycles is an important consideration when you program the vector sequencer, and this subject is covered in the manual *Advanced Testing with the HP 82000.*

Sequencer Control Window

The Sequencer Control Window allows you to program the sequencer more easily than by using the Sequencer Programming Window. Each line of the program allows you to send a block of vectors to the DUT, and allows you to repeat these vectors indefinitely, or until a fail condition is detected. In each line of the Sequencer Control Window you can do the following:

- cycle a block of vectors and specify different stop conditions

- execute a block of initializing vectors before the test vectors.

Figure 6-14 shows a typical setup in the Sequencer Control Window. The first line in this window always shows you the full range of the vectors present in the vector memory. This line has the label DEFAULT. It is automatically inserted whenever you enter vectors in the Vector table.

Figure 6-14. Sequencer Control Window

You can program up to 15 lines in this window. Each
line represents a complete functional test. The data you
can program for each line is described below.

Sequencer Labels

The label field is used to identify a particular test
sequence. You should label each line with a descriptive
name. It is this name that you enter in the Test Control
Window when you want to run one of the tests in a
series. Define where you want the sequencer to start.
When you start the sequencer from the Test Control
Window, you need to specify a label as the starting
point for the sequence. When you open the Test Control
Window, you enter the test label in the edit box to run
a sequencer test. The label must be an alphanumeric
string.

Executing a Single Block of Vectors Once

The simplest instruction that you can give in this
window is to send a single block of vectors to the DUT.

The sequencer starts at the address specified in the
first test vector field and sequentially executes the
subsequent vectors, up to and including the last vector
address (specified in the last test vector field. The
sequencer then leaves the system in the break state,
where the break vector is sent to the DUT.
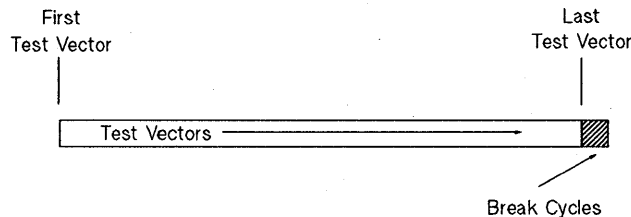
Figure 6-15. Executing a Single Block of Vectors Once

Initializing the DUT Before Testing

With some devices, you may find it necessary to initialize your DUT before performing your tests (especially if you are testing sequential logic elements). You can do this from the Sequencer Control window, by specifying a block of vectors to be executed only once, *before* your sequence of test vectors is executed.

The initialization block starts at the vector number that you enter in the `initialization start vector` field and finishes at the first test vector.

If you specify an initialization block, the results are *analyzed* starting from the first address of the test vector block (not the initialization block). The first *test* vector is numbered as test cycle 0 in the Test Result windows (Error Map, State List and Timing Diagram). The initialization vectors are numbered as negative test cycles.
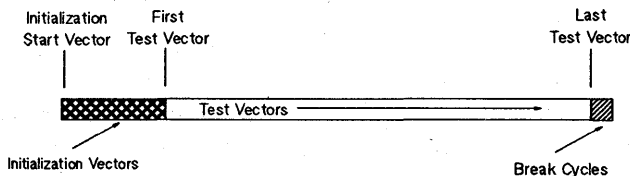
Figure 6-16. Executing a Single Block of Vectors with an Initialization Block

Cycling a Block of Vectors

If the `repeat until` field is left blank, the vector block will be executed once, and the sequencer then enters the break state.

Break on Fail

You can instruct the sequencer to repeatedly execute a block of vectors until a fail condition occurs, by entering `fail` in the `repeat until` field. You can, as mentioned above, also use one or several vectors at the start of this block to initialize the DUT. The vector block that you

use for testing (the block that is continuously executed), starts immediately after the initialization block. The initialization block is only executed once.

The sequencer repeatedly executes the specified block until a failure is detected. The sequencer enters the break state at the failed vector, allowing you to examine the failure with the Test Result windows.
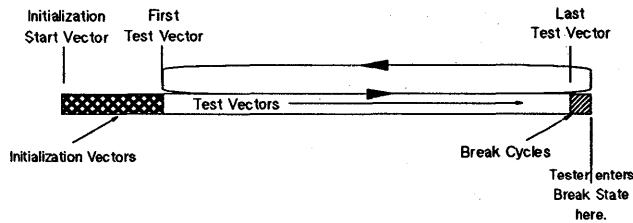
Figure 6-17. Repeating a Block of Vectors

Cycling a Block of Vectors Indefinitely

You can specify that a block of vectors is indefinitely executed, by entering `infinitely` in the `repeat until` field. In this case, failures are ignored by the tester and it does not enter the break state automatically.

7

Jsing the Standard Test Functions

This chapter describes the use of the pre-written test functions, from the Test Control window. You can find detailed descriptions of the tests themselves, and information about the input parameters of the test functions, in the manual *Standard Test Function Reference*. Information about linking tests together into programmed test flows is described in *Advanced Testing with the HP 82000*.

You can also adapt these standard test functions to your own specific testing requirements, or even write your own test functions. The manual *Test Function Programming for the HP 82000* gives you a comprehensive guide to this subject.

Standard Test Function Groups

Standard test functions that are supplied with the system are categorized into the following groups;

DC Tests

- **continuity** - checks that the DUT is connected to the tester before testing commences. It also tests chip bonding and protection circuits, and checks for short circuits on the DUT board.
- **dvm** - measures the voltage at any pin.
- **high_z** - measures the leakage current of outputs in tristate condition.

- **input_dc** - measures the dc input characteristics of the DUT.
- **operating_current** - measures the active operating current of the DUT.
- **output_dc** - measures DUT output voltage.
- **standby_current** - measure the standby current of the DUT.

AC Tests

- **acquire_data** - a functional test in data acquisition (DA) mode.
- **error_count** - a functional test reports back the number of errors detected. Test results can be displayed in the Error Map.
- **funct_reload** - a functional test giving a pass/fail indication and allowing vector reload without stopping the system clock.
- **functional** - a functional test in real time compare mode (RTC), giving a pass/fail indication. The test results can be displayed in the Error Map.
- **global_search** - varies a single measurement parameter until a Pass/Fail or Fail/Pass transition is detected.
- **hold_time** - measures the data hold time.
- **inp_volt_sensitivity** - performs a functional test while varying the input levels of the DUT.
- **out_volt_sensitivity** - performs a functional test while measuring the high and low output levels of the DUT.
- **propagation_delay** - measures the propagation delay of the DUT.
- **rise_fall_time** - measures the rise and fall transition times of the outputs.
- **setup_time** - measures the minimum time that the (signals must be setup before a clock transition.

Sweep Tests

- **shmoo_2d** - performs a two dimensional shmoo plot.

Other Tests

- **comment** - adds a comment line to the results output.
- **get_utility_lines** - reads the state of the utility line inputs.
- **golden_device** - reads the output data from a DUT test and enters (stores) it into the vector table as expected data.
- **header** - defines a header to be used for test results.
- **set_utility_lines** - sets the utility line outputs.
- **temperature** - reads the temperature inside the mainframe; for calibration purposes.

In addition to executing these tests from the Test Control window, there are basic control functions that help you manage the tests. You can select and execute these control functions from the Test Control window in the same way as you would for the test functions.

Common control operations are listed below:

Control Test

- **connect** - connects DUT to tester resources
- **disconnect** - disconnects DUT from tester resources

Sequencer Control

- **start_sequencer** - starts and stops the vector sequencer
- **test_state** - gets the present state of the tester (running, off or break) and the status of the PASS/FAIL flag.

Download/Upload (Transfer Setup Files)

- **download** - loads setup files into the tester hardware.
- **upload** - reads tester hardware settings and stores them in the workstation as setup files.

Comparison Modes

Each receiver channel of the HP 82000 can be set in one of two comparison modes: Real Time Compare (RTC) mode and Data Acquisition (DA) mode. The compariso mode is set by the firmware of the tester, depending on the type of test function you have chosen in the Test Control window.

Real Time Compare (RTC) Mode

In this mode, the expected data is downloaded to the pins' receive pattern memory before a test commences. If you specified that a block of initializing vectors is downloaded to the DUT, comparison begins when the first vector in the following block is downloaded. When comparison begins, the pin output is sampled within a window or at a clock edge, depending on the pin's configuration (refer to the chapter "Configuring Pins for Testing").

The expected data is read out of the receive pattern memory at the same rate that data is received from the DUT. The DUT levels are compared with the levels set in the Level Setup Window, and a high, low or intermediate value is registered accordingly. This result is then compared with the expected result from the pattern memory. If the signal level deviates from the expected level, a failure for that cycle is registered. The failures for all pins are ORed to give a failure status for that test vector. The failure status for all vectors is stored in an Error Map memory. The contents of this memory is retrieved by the software and analyzed to giv the test results.

The results for all pins are ORed together in the Error Map, to give an error indication to vector level. The Pin Error List additionally identifies the DUT pins which have failed.

To localize a failure to bit level, you can then perform a more detailed analysis in the Data Acquisition Mode.

Figure 7-1 shows how the receiver side of an I/O channel is configured, while testing in RTC mode.
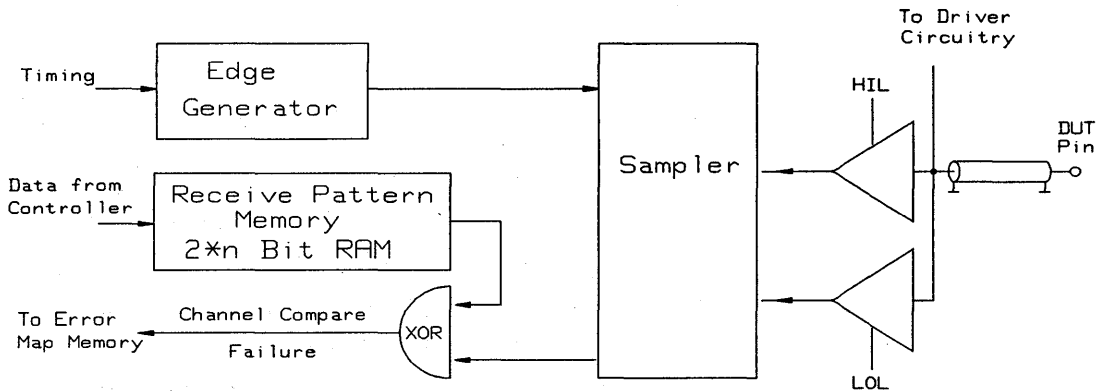
Figure 7-1. RTC Circuitry - Simplified Block Diagram

Data Acquisition (DA) Mode

No data from the controller is stored in the Receiver Pattern Memory. The signals from the DUT are compared with a high and low threshold, which are set in the Level Setup Window. As with the RTC Mode, the type of strobe can be either edge or window. The sampling edges are individually programmable for each pin (refer to the chapter on Timing Setup) and each comparison can have one of four results. These results are encoded in two bits as shown in Table 7-1 below:

Table 7-1. Data Acquisition Results

Bit 1	Bit 0	Meaning
1	0	High (higher than high)
0	1	Low (lower than low)
0	0	Intermediate (high > signal > low)
1	1	Unstable (for window compare only)

These comparison results are transferred to the
controller where they are compared with the expected
data. The expected data in the controller is structured
similarly to the received data from the tester and has
two bits allocated per cycle. These encode the following
four possibilities that you can enter:

i   The level is expected to be an intermediate level.

X   Don't care. The result is always PASS.

1   The level should be above the high threshold set
        in the Level Setup Window.

0   The level should be below the low threshold set i
        the Level Setup Window.

Figure 7-2 shows how the receiver side of an I/O channe
is configured, while testing in DA mode.
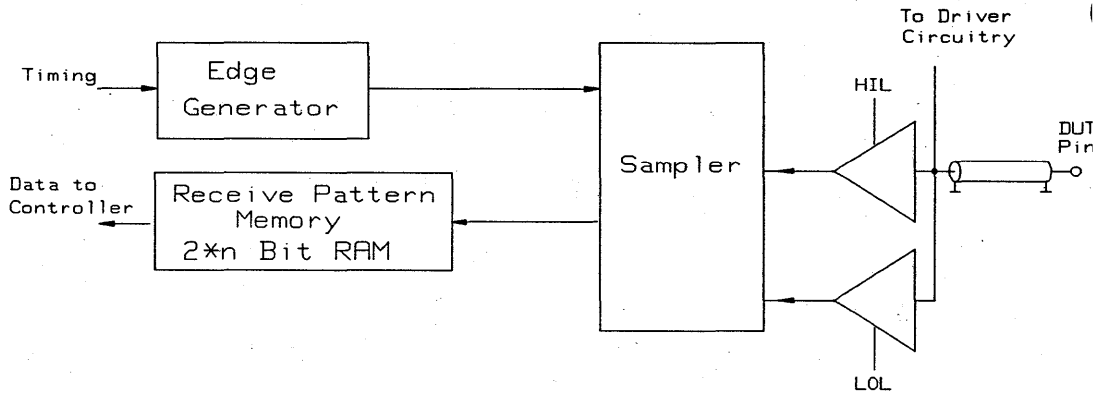
Figure 7-2. DA Circuitry - Simplified Block Diagram

Selecting a Test

)

You can select a test function from the Test Control window by selecting the new option from the File pulldown menu.

The test functions are divided under the following group headings:

- DC Tests
- AC Tests
- Sweep Tests
- Other Tests
- Test Control
- Sequencer Control
- Download/Upload
- Miscellaneous Tests

The group *AC Tests* is displayed by default when you select the new option. You can choose a different group using the fill menu in the group entry field.

Once you have selected the correct group, all of the tests within that group are displayed in the browser. You can select the test you want by clicking the browser entry in the usual way, and then clicking the (new) pushbutton.

If you do this for the functional test, the screen for this test function will be displayed as shown in Figure 7-3. The window layout for this test is quite simple, because there is only one parameter to enter; the sequencer label from which you want the test to start. The sequencer labels are defined when you set up the vectors for testing in the Sequencer Control window.

You can find a detailed description of the built-in test functions, the parameters, and how to use them, in the manual *Standard Test Function Reference*.
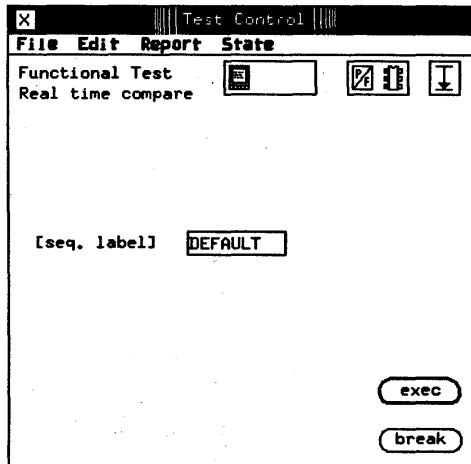
Figure 7-3. Functional Test - test function

Setting the Test Parameters

The field seq label is automatically set to a label called DEFAULT. This label is always set as the first instruction in the Sequencer Control window, and defines a linear sequence starting from vector 0 and finishing at the vector before **ffvad** (the first free vector address).

When you are satisfied that you have selected the parameters you need, you can execute the test by clicking the exec pushbutton.

Saving Test Parameters

If you want to use this particular test often, you can save it in the testfunc subdirectory of the device you are testing.

To do this, select the save_as option from the File pulldown menu. The system responds by displaying the save test as dialog box shown in Figure 7-4. You can now enter a name for the test and save it by clicking the save_as pushbutton.

```
┌─────────────────────────────────────────┐
│ ┌───────────────────────────────────────┐ │
│ │                                       │ │
│ │  save test as          ( save_as )    │ │
│ │  ┌─────────────────┐                  │ │
│ │  │ rubens_demo█    │   ( cancel  )    │ │
│ │  └─────────────────┘                  │ │
│ │                                       │ │
│ └───────────────────────────────────────┘ │
└─────────────────────────────────────────┘
```
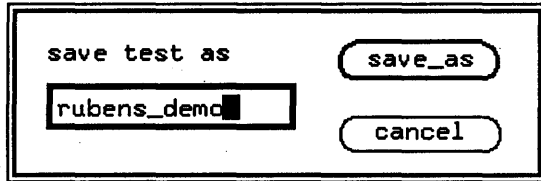
Figure 7-4. Saving Your Test

If you save the test, it will be stored in the testfunc
subdirectory of your device. The file that is saved has
the name that you gave to your test. The file itself
contains the name of the test function and the list of
parameters for that function.

Storing and Recalling Tests

As described above, you develop tests by calling up a test function, altering the parameters to define the test, and then storing the test and parameters under a name of your choice. When you want to recall a test that you previously set up, you select the load option on the File pulldown menu. The load test dialog box is then displayed as shown in Figure 7-5.


```
┌─────────────────────────────────────────┐
│ ┌───────────────────────────────────┐   │
│ │  device pathname                  │   │
│ │  ┌─────────────────────────────┐  │   │
│ │  │..                           │  │   │
│ │  └─────────────────────────────┘  │   │
│ │  ┌──────────────────┐             │   │
│ │  │dc_operating_c  ▲ │             │   │
│ │  │dc_output_volt    │             │   │
│ │  │dc_standby_c      │             │   │
│ │  │dl_d200           │             │   │
│ │  │dl_d50            │             │   │
│ │  │dl_last_sett      │             │   │
│ │  │dl_mux_d200       │             │   │
│ │  │dl_mux_d50        │             │   │
│ │  │ot_comment        │             │   │
│ │  │ot_golden_dev     │             │   │
│ │  │ot_header         │             │   │
│ │  │ot_temperature    │             │   │
│ │  │rubens_demo       │             │   │
│ │  │sc_start_seq      │             │   │
│ │  │st_pulse_mux      │             │   │
│ │  │st_pulse_mux50    │             │   │
│ │  │st_shmoo_V_mux    │             │   │
│ │  │st_shmoo_Vee    ▼ │             │   │
│ │  └──────────────────┘   ( load )  │   │
│ │  load test                        │   │
│ │  ┌──────────────┐     ( cancel )  │   │
│ │  └──────────────┘                 │   │
│ └───────────────────────────────────┘   │
└─────────────────────────────────────────┘
```
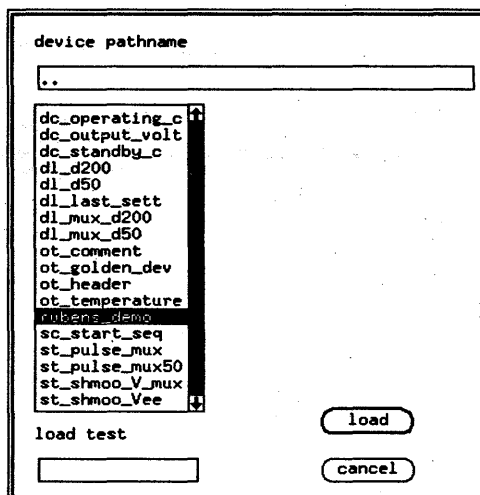
Figure 7-5. Loading a Pre-saved Test

All of the files stored in the testfunc directory of your device are displayed in the browser. If you saved the functional test in the example above, you will see it also listed. You can also load test files from another device, although you must be sure that the configuration of the device is the same. You need to enter the full path to the other device in the device pathname edit box. The browser will then contain the tests that were set up for that device.

Running a Test

Once you have selected your test and filled in the necessary parameters, you have several possibilities for running it, depending on your application. When you run a test, you can set the following modes for testing:

Report mode   sets a destination for the report information. This can be the report window, a file, a printer, or any combination of the three.

Test mode   selects the type of result to be returned by the test. This can be simply a pass/fail indication, or a measured value. Results can be returned for the pin list as a whole, or on a pin-by-pin basis.

Repeat mode   determines how often the test should be repeated. This can be just once, until a failure occurs, or endlessly.

The status of these three modes for testing is shown in the icon field at the top right of the Test Control Window (refer to Figure 7-6).

Figure 7-6. The Report, Test, and Repeat Mode Icons

Setting the Report Mode

When you want to set the report mode, click the left icon box in the Test Control Window. The report mode dialog box is then displayed as shown in Figure 7-7. To select an option, all you need to do is click the appropriate icon. The icon is then *boxed* and appears in the icon field at the top right of the Test Control Window. This method of selecting options is the same for the Test mode and Repeat mode dialog boxes.
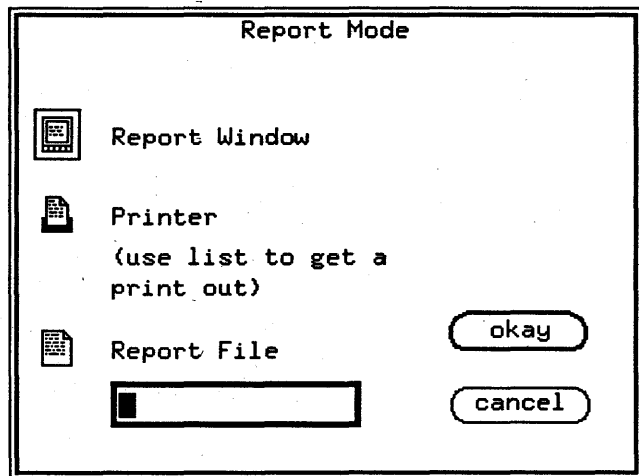
Figure 7-7. Setting the Report Mode

The Report Window Icon

When you select this option, error and status messages are displayed in the Report Window. This is the default report mode that the software has when first run.

The Printer Icon

All report messages will be routed to your local printer.

The Report File Icon

Report messages will be routed to a file of your choice. You need to specify the file name in the Report file edit box. The system creates a file with this name in the reports subdirectory of the device you are using.

Setting the Repeat Mode

Click the far right icon box on the Test Control Window to set the repeat mode. The Repeat Mode dialog box is then displayed as shown in Figure 7-8. The repeat mode dictates the end condition for the test you are running. You can run a test just once, until a failure occurs, or infinitely. This allows you to vary external conditions (such as power, temperature, and so on) and observe their effects on the test results.

```
┌─────────────────────────────────────────────┐
│╔═══════════════════════════════════════════╗ │
│║              Repeat Mode                  ║ │
│║                                           ║ │
│║  ┌───┐                                    ║ │
│║  │ ⊥ │   repeat once                      ║ │
│║  └───┘                                    ║ │
│║                                 ╭────────╮║ │
│║   ⟳     repeat endless          │  okay  │║ │
│║                                 ╰────────╯║ │
│║                                 ╭────────╮║ │
│║   ⟳     repeat until failed     │ cancel │║ │
│║                                 ╰────────╯║ │
│╚═══════════════════════════════════════════╝ │
└─────────────────────────────────────────────┘
```
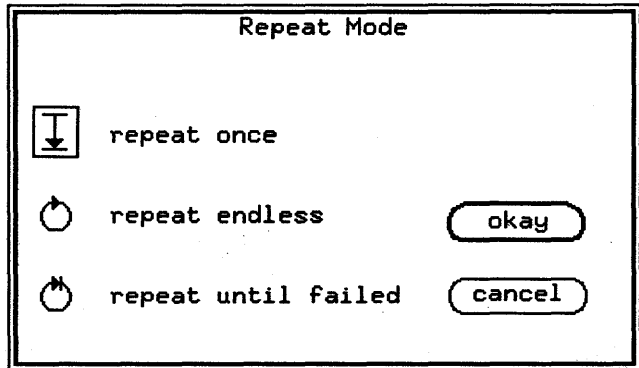
Figure 7-8. Setting the Repeat Mode

The Repeat Once Icon

$$\underline{\overline{\mathsf{I}}}$$

The test will only run once. This is the default test
mode, set when the software first runs.

The Repeat Until Failed Icon

$$\circlearrowright$$

The test will be repeated until the first failure occurs.
This allows you to vary external conditions until the
device functionality breaks down. You can then examine
the results and analyze the failure.

The Repeat Endless Icon

The test vectors will be repeated endlessly, in an
unconditional loop, regardless of the test results. To stop
the test, you need to click the (break) button on the Test
Control Window (refer to Figure 7-3).

Setting the Test Mode   The test mode selects the scope of the results that will be returned by the test. Naturally, this also depends on the type of test you are running, and you need to consider the test mode in terms of your application.

When you click the middle icon box, (refer to Figure 7-6), the Test Mode dialog box is displayed as shown in Figure 7-9. To select a different test mode options, click the appropriate icon. The test mode consists of two parameters; the test method, and the test evaluation.
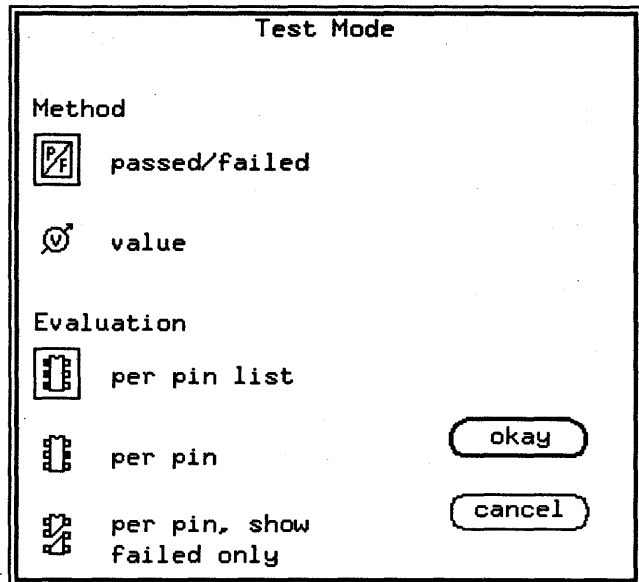
Figure 7-9. Setting the Test Mode

Setting the Test Method

The test method determines what type of result will be returned for the pin(s). You can select whether a simple pass/fail indication will be returned, or a measured value.

The Pass/Fail Icon

The test will return only a pass/fail indication for the pin(s). For further information on the different result formats available, refer to the relevant results window chapter.

The Value Icon

The test will return a measured value for the pin(s) as well as a pass/fail indication.

Setting the Type of Evaluation

You can also select whether a failure indication or evaluation is to be provided on a per pin, or pin list basis.

The Pin List Icon

Measurements are made on all pins in the defined pin list, and a single result is output.

The Per Pin Icon

Measurements are made on each pin separately and a separate result for each pin is output.

The Per Pin, Show Failed Only Icon

Measurements are made on each pin separately, but only the results for the *failed pins* are output. If all pins pass, a result range is output, similar to the pin list output.

Performing Functional Tests

The first four tests listed in the AC Tests group, allow you to functionally test the DUT. These tests do not change any timing or level parameters during evaluation, but only exercise the DUT at nominal settings specified in the level and timing setup windows.

The first step in your test plan should always prove that the device (DUT) is functional. You can then apply more specific AC tests to your DUT, and gather test data and retain it if necessary.

The results information generated by the functional tests are as follows:

Basic Functional Test[1]	Returns *Pass/Fail* for complete test
Functional Test with Acquire Data[2]	Returns *DONE* and the test results can be analyzed using the timing diagram or state list.
Functional Test with Error Count[2]	Returns *Pass/Fail* with total number of failed vectors

[1] Error Map
[2] Timing Diagram and State List

Automatic reloading of vectors is not supported by the test functions. However, a number of vector setups can be loaded consecutively using the *funct_reload* test function.

Terminating Tests

When the repeat option is used to perform an endless test, the tester hardware resources are blocked for any other tasks, until the test is terminated by the (break) pushbutton.

<table>
<tr><td>

Caution 🖐

</td><td>

If you try to use the (break) pushbutton to terminate an endless loop in a sequencer program, the tester hardware may be left in an unstable state, and this can cause the system software to shutdown.

Avoid this situation by using the **sequencer_start** test control window when you run sequencer loop-programs. You can find details about programming the sequencer, in the manual *Advanced Testing with the HP 82000*

</td></tr>
</table>

When the *repeat until failed* option is selected from the test mode window, and it detects an error, then the tests will also terminate.

If you need access to other tester resources while the sequencer is running (for example, for scope measurements), use the **sequencer_start** test function.

DC Testing with the PMU

Most of the test functions in the DC Tests group make their measurements using a Parametric Measurement Unit (PMU) and in order to interpret the results of these test functions, it is important to understand how the PMU operates.

Modes of Operation

You can use the PMU to force a voltage or a current at any pin of your DUT, and to measure the current and/or voltage. In order to protect the DUT from a power overload, the PMU allows you to set a limit for either the current or voltage applied during the test.

The PMU has four modes of operation:

- for forcing a voltage (a *voltage source*)
- for forcing a current (a *current source/sink*)
- for making voltage measurement (a voltmeter)
- for making current measurement (an ammeter)

The mode in which the PMU operates is dependent on the following factors:

- the loading conditions of the DUT pin;
- the PMU parameters that you have set.

A Passive Loading Example

The following example shows how the mode of operation varies under different load conditions.

The PMU is programmed to the following values:

- Force voltage = 10 V
- Positive current limit = 20 mA
- Negative current limit = −20 mA

Figure 7-10 illustrates the the operating characteristics of the PMU for different values of load resistance (R_L).
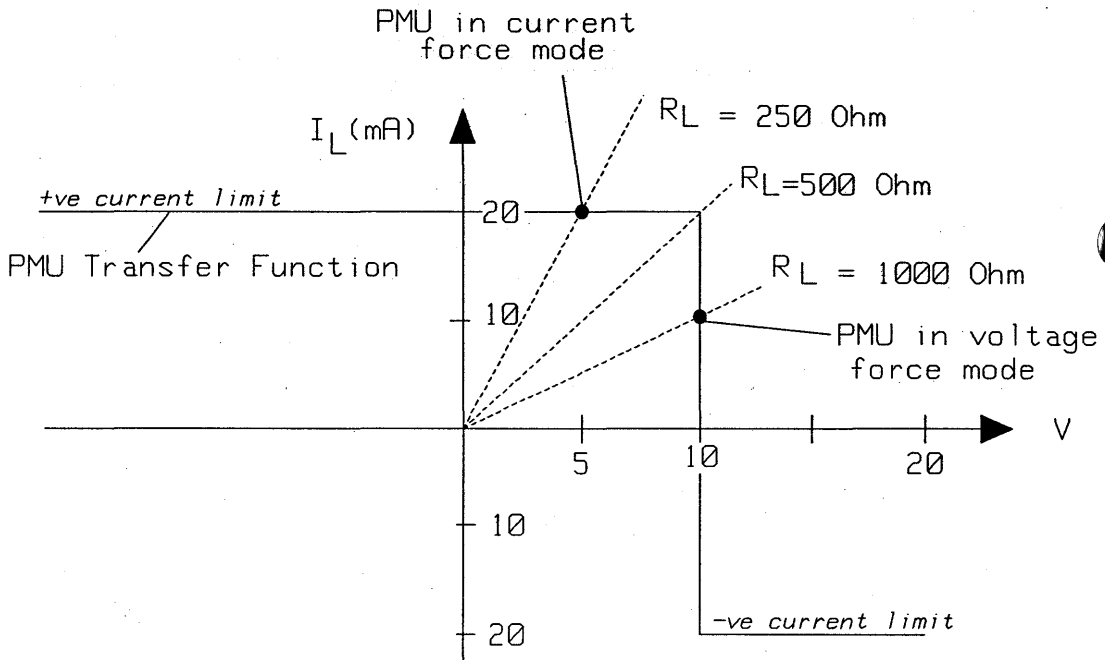
Figure 7-10. PMU Operation under Varying Passive Loads

From Figure 7-10, you can see that:

- if $R_L > 500\ \Omega$, the PMU operates as a **voltage source**;
- if $R_L = 500\ \Omega$, the PMU output reaches its maximum allowable current (20 mA);
- if $R_L < 500\ \Omega$, the PMU output can not reach the force voltage value (10 V) without exceeding the maximum allowable current. So, the output voltage is controlled, and the PMU then operates as a **current source.**

The PMU works in the same way for a negative-polarity force voltage, except that the operational area shifts 180° from the first quadrant to the third quadrant of the graph. A load *less* than 500 Ω would then activate the negative current limit.

Summary

The operating mode of the PMU is dependent on the external load.

An Active Loading Example

The following example shows how the PMU performs an output dc voltage measurement under current load conditions of 20mA. The PMU acts as a current sink in this case.

The DUT has the following specification:

- Load Resistance $R_L = 100\Omega$
- Output Voltage $V_{out} = 4$ V
- Load Current $I_L = 20$ mA

The PMU is set up with:

- Force Voltage $= 0.5$ V
- Neg. current limit $= -20$ mA
- Pos. current limit $= 20$ mA

Figure 7-11 illustrates this test set up.

PMU

DUT

PMU output voltage = 2 V   I = -20 mA

U = 1,5 V

75 Ohm

U = 0,5 V

U = 2V

100 Ohm

U = 4 V
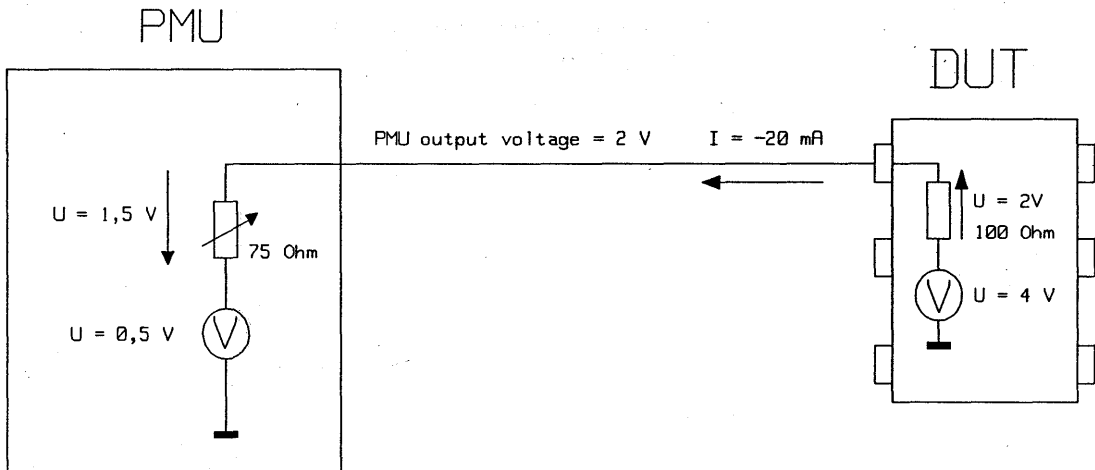
Figure 7-11. Output DC Voltage Measurement with the PMU

At the programmed PMU force voltage (0.5 V), the total potential difference across the 100 Ω load resistance is:

4 − 0.5 = 3.5 V

In this situation, the PMU would have to sink a current of 35 mA. However, the *negative* current limit is set to −20 mA, and so the the PMU output voltage is increased to +2 V to keep the current at this level. This is shown by operating point **P1** in Figure 7-12.

Changing Mode by Varying the External Load

In the test set up in Figure 7-11, the output load is decreased so that the output load characteristic intersects 0V. This means that the PMU operates as a *voltage source* with

- 0.5 V output voltage
- and +5 mA output current (note the sign change).

This is shown graphically as operating point **P2** in Figure 7-12.
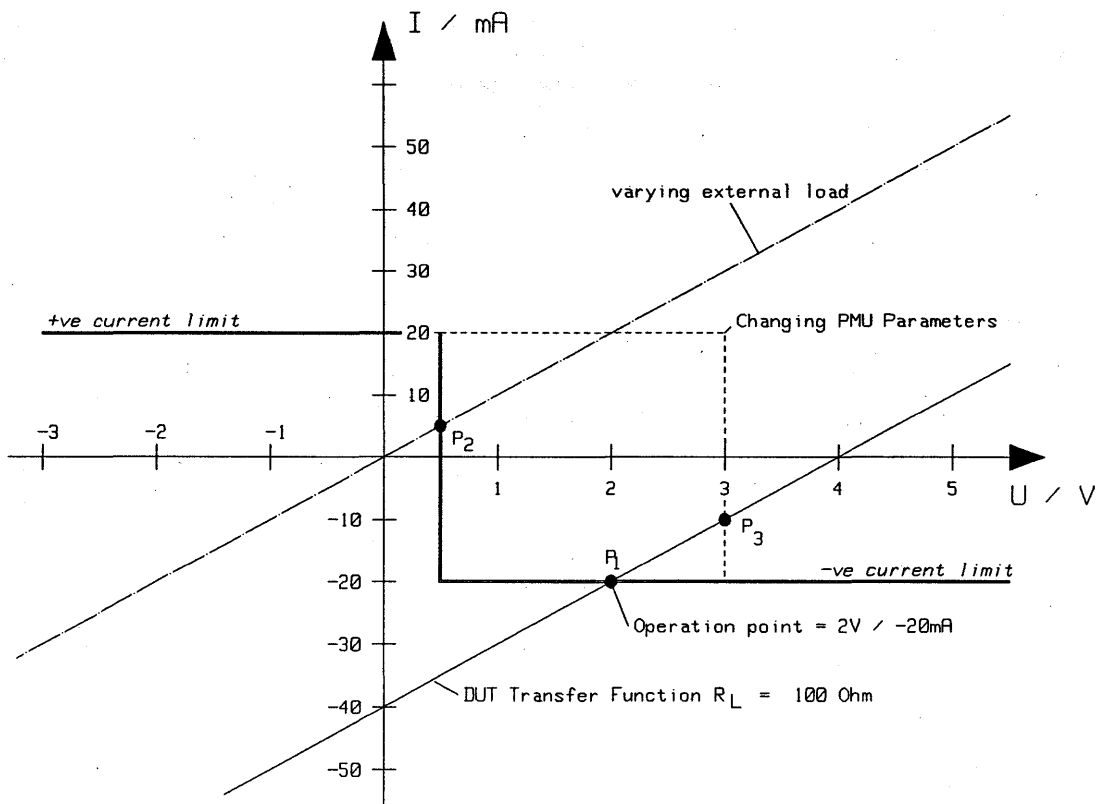
Figure 7-12. PMU Operation under Varying Active Loads

Changing Mode by Changing PMU Parameters

If the PMU force voltage in Figure 7-11 is then increased from +0.5 V to 3 V, this also changes the mode of operation of the PMU. The PMU then operates in Voltage Force mode, with an output current of -10 mA. This is illustrated in Figure 7-12 as the operating point **P3.**

In either of the last two cases (changing the PMU or the external load), when the current load is less than 20 mA, the DUT output voltage measured is the voltage being forced by the PMU.

Summary

1. The PMU mode depends on the external load.

2. The PMU mode depends on the settings of the PMU parameters.

Setting Up a PMU Test Function

Figure 7-13 shows how to use the a test function window to define the PMU parameters used in the previous example.

The example measures the DC output voltage of the DUT, under a defined current load, and the test function which is used for DC output measurements is called *output_dc* listed in the DC Tests group.
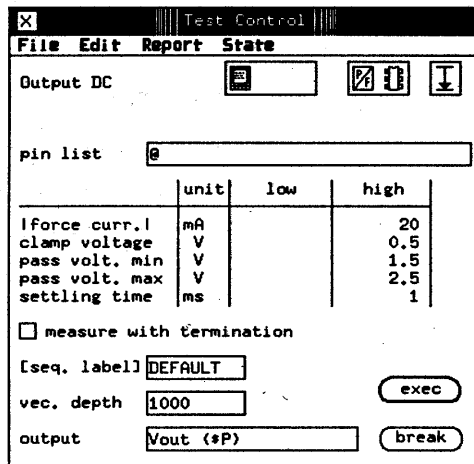
Figure 7-13. Test Window

How the PMU Circuitry Operates

Forcing a Voltage

The PMU is set to a constant output voltage until the current through the output load reaches either the upper or lower current limits. The bold lines in Figure 7-14 show the signal path.
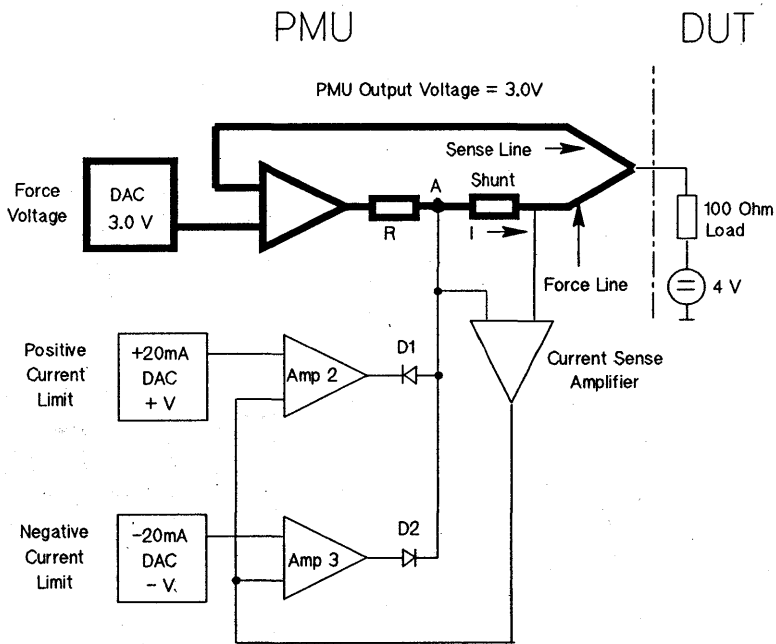
Figure 7-14. PMU in Voltage Force Mode

If the output load is high enough to keep the output current *less* than the current limit value, diodes D1 and D2 are open, because Amp 2 is at its highest positive level and the output voltage of Amp 3 is at the lowest negative level. Only the voltage force mode is active.

The PMU output voltage is equal to the programmed force voltage.

Forcing a Current

As the output load decreases, the current through the shunt and the external load increases, and so does the voltage across the shunt resistor. This increase of current is detected by the Current Sense amplifier. The signal path is shown in Figure 7-15.
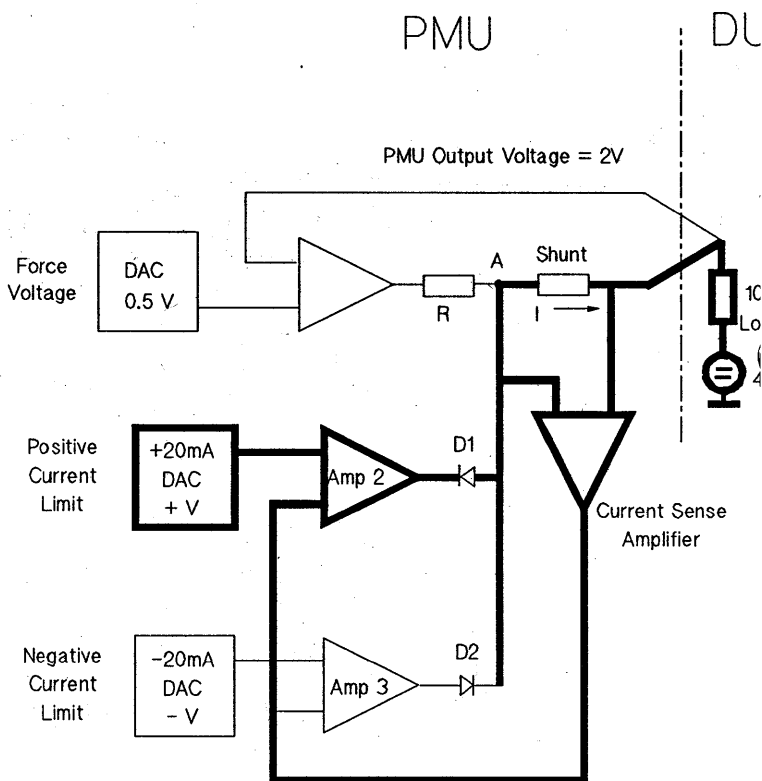
Figure 7-15. PMU in current force mode

When the output from the Current Sense amplifier reaches the same value as either of the current limits that you have set, the positive or negative current limit diode closes (depending on direction of the current). The

current amplifier then supplies a voltage at point **A**, in order to maintain the current flowing at the programmed limit.

Under these circumstances the PMU output voltage is no longer equal to the programmed voltage.

Setting the PMU Modes

Setting the Current Direction

The direction of current flow is determined by whether you set the PMU force voltage above or below the external source voltage.

Note 👆 Positive currents flow into the DUT pins, and negative currents flow out of the DUT pins.

Setting the PMU as a Current Source or Sink

Set up the PMU as follows:

■ Specify the current limit

■ Specify current direction by programming the PMU force voltage.

In the current source/sink mode, the PMU force voltage is called the "clamp voltage". As the current decreases, a current source/sink increases the output voltage in order to keep the current constant. The PMU only allows the voltage to increase to the force voltage level, as programmed. When that level is reached, the PMU switches from current force to voltage force, to "clamp" the PMU output voltage to that level.

■ Specify a PMU voltage ("clamp voltage") that will allow more current to flow than the specified current limit. This way the PMU is always able to maintain the current at the limit value and therefore works as a current sink/source.

Setting the PMU as a Voltage Source

- Set the PMU voltage force level

- Set the current limit higher than the current which
  you expect to flow (but do not exceed the current
  limits of your device!)

Summary

To determine the state in which the PMU is working,
you can apply the following rules:

1. Check the measurement results in the report screen

2. If the result voltage is identical to the programmed
   PMU clamp voltage, then the PMU is in force voltage
   mode

3. If the result current is exactly the value specified as
   the current limit, then the PMU is working in current
   sink/source mode

Result Display Interactions

This section describes the interaction of the test
functions with the Result Displays (Error Map, State
List, and Timing Diagram).

Real Time Compare Mode

Test functions running in this mode:

- Block the Timing Diagram and State List displays -
  since they are not valid for RTC mode.
- Update the Error Map at the end of a test run.

The results in the Error Map are normally only useful
after running a functional test, a functional reload or
functional test with error count. Other tests performed
in RTC mode, such as propagation delay, perform several
functional tests with timing or level changes. The Error

Map and Pin Error List displays only show the results of the last test.

Data Acquisition Mode

Test functions running in this mode:

- Block the Error Map - since no expected data is available
- Update the Timing Diagram and State List displays at the end of a test run

The results in the Timing Diagram and State List displays are normally only useful after running a functional test with acquire data. Most of the other test functions perform a functional test in RTC mode, and the results of the last test are available at the end of the test run.

Exceptions

There are a few exceptions to the above mentioned interactions.

The result displays are also disabled during:

Download	If relevant setup data is modified
Shmoo	To speed up this test function
DC Tests using Sequencer	No relevant data available

The result displays are not disabled for:

DC Tests without Sequencer	No change in previous result data

Vector Reloading

Note that all test functions that start the sequencer during execution will stop the system clock before starting the test to reset receiver resources and capabilities.

The only exception to this is the funct_reload test, which can be used for vector setups that are longer than the system normally allows. Note that each vector setup must be a complete setup with its own sequencer instructions. See the description of the funct_reload test function, in the *Standard Test Function Reference* manual, for more details

General Information

Report Window Result Accuracy

The test results displayed on the Report window are rounded to three significant digits to save space in the display. The test function result PASSED or FAILED is generated by measuring the test result to its full accuracy. This can sometimes lead to inconsistencies in the displayed value when the measured value is close to the pass value.

The following example illustrates this point.

```
        3590 mV                        4850 mV
           |                              |
  FAIL  |-------------- PASS -------------->|   FAIL
           |                              |
 _____
          ↑↑                            ↑↑

      3588 - Display = 3.59V       4852 - Display = 4.85V
             Status = Failed              Status = Failed
```
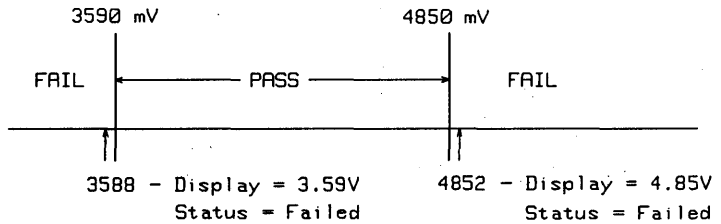
Figure 7-16. Measurement Accuracy Example

Although the measured value was 3588 mV, the displayed value will be rounded up to 3.59 V (more than the lower limit), but the test function status will indicate that the test failed.

When you are using the test functions from a testflow program in BASIC or C, the test results are returned with full accuracy, in this case, 3588 mV.

Operating Mode   All AC Tests are performed with the system in Real Time Compare mode, except for the Acquire Data test, which is performed in Data Acquisition mode.

Tester State   Unless stated otherwise, all AC Tests set the tester to the CONNECTED state when they finish execution.

Level and Timing Setup Considerations   When you perform an AC Test that gives a result value, the test should be performed using nominal setup values for the levels and timing margins. To ensure a valid evaluation of the timing or level parameters, the device must provide a *pass* result for a functional test.

The parameter that is to be evaluated further, is then swept from the nominal setting (that you have specified in the setup window), until a *pass/fail* transition is detected.

Timing Measurements in the Next Cycle

If you perform a timing characterization test which causes a measurement to be made in the *next* period, there is an additional measurement uncertainty of ±0.1% caused by the system period inaccuracy.

DUTs Requiring Initialization

If a DUT requires an initialization sequence, the sequencer program must have the following form:
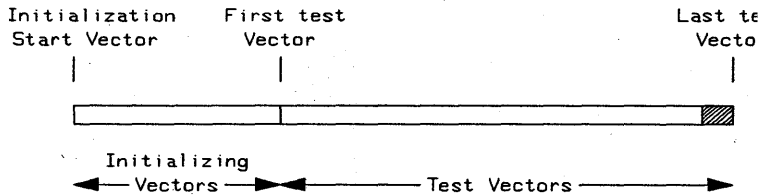
Figure 7-17.
Sequencer Program with Initialization

The measurement pin(s) must not be used for the initialization sequence jump condition. This is because the timing is constantly modified for these pins throughout the whole test.

Note

Some operating modes do not allow individual pin masking for the anticipated data during the initialization sequence. Consequently, you must mask all DUT outputs using the wordmask.

Parallel/Serial Testing   When there is a parallel/serial radio button shown in the test function window, the button setting will override the Test Options setting for that test, should conflicts arise.

For tests without this button, the following rules apply:

Test Mode	Comment
pin list	Measured value (or output to file) and one state (*pass/failed*) is displayed for complete test
per pin	Value(s) and state are displayed for each pin

For tests with a serial/parallel button, the following table shows the results generated.

Evaluation	Test Method	
	Serial	**Parallel**
Per Pin	1 value per pin	1 or 2* values for list
	1 state per pin	1 state for list
Pin List	2 values (min, max) for list	1 value for list
	1 state for list	1 state for list

* Two values are displayed for the rise/fall time test function only, as this is always measured per-pin.

Reference Pin Specification   The pin that is defined as a reference pin for an AC test, must always be an input pin carrying a tester drive signal.

Sequencer Label

The DEFAULT sequencer label, which specifies a linear sequence from the first vector address to the last used vector address, is the system default recommendation. You may change the label to another that you have previously defined in vector sequencer control programming for specific test requirements.

If no label is specified, the tester resumes operation from the last sequencer state the system was at, prior to calling this test function. If that is a break state, the tester may, or may not resume at the DEFAULT label, depending on what has been programmed.

When the test system has been disconnected, it will star at the DEFAULT label.

DUT Input Stability During DC Tests

The DUT inputs MUST be in a stable condition while DC measurements are being made.

Remember that if you have pins with the signal formats

- RI
- RC

the signals will *still* toggle while the sequencer is in the BREAK state. Therefore, if pins with these signal formats have an effect on the stability of the DUT, avoid using these formats during DC measurements.

RZ and R1 format pins may also affect the stability of your device. To prevent this, the break vectors for:

- RZ pins must be set to 0;
- R1 pins must be set to 1.

For this reason, if the break vector for RZ or R1 format pins is set to hold while you perform a dc measurement, the pin may be held in a toggling state.

Using the Error Map

What the Error Map Does

The Error Map window allows you to see the results of functional tests carried out in the Real Time Compare (RTC) Mode. In RTC Mode, the *actual* data received is compared in the tester hardware against *expected* data, stored in the receiver pattern memory.

The Error Map illustrates all the test cycles, in a matrix of dots, and indicates any discrepancies between the expected and received vectors.

The Pin Error List, which is a sub-window of the Error Map gives an overview of all output pins (tester receive pins) and flags those pins where one or more errors have occurred in the whole of the vector range.

Results of tests performed in the Data Acquisition (DA) Mode appear in the Timing Diagram and State List Windows, and do not affect the Error Map display.

Activating the Error Map Window

To activate the Error Map click the key (Error Map) in the Main Menu Bar. The Error Map Window appears in the lower left hand corner of the screen.

As Figure 8-1 shows, the window is divided into two areas, a left hand area which gives various values relating to the Error Map display, and a right hand area which shows the result of the functional test.

If, after opening the Error Map Window, no data for the Error Map is available, the result area of the window comes up empty (this is the case in Figure 8-1). A warning message in the Report window appears:

```
Data for Error Map not available. Run test
function "functional".
```
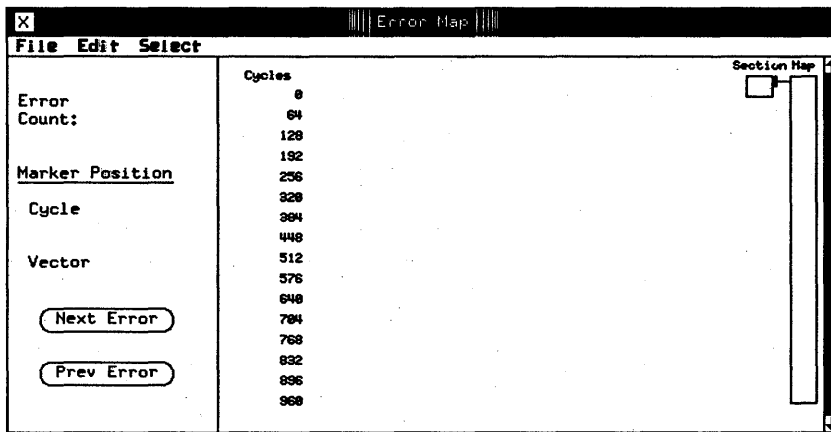

Figure 8-1. Error Map Window - No Error Data

Checking for Errors in the Full Cycle Range

The result area gives an overview of errors that have occurred anywhere within the test. The overview is provided by four elements:

- a passed/failed message at the top of the result area
- the Map column, on the right of the result area
- the Section column, to the left of Map
- one Error Map *page,* in the center of the result area.

Map Column

After executing a test function in Real Time Compare mode (such as `functional`), the **Map** column contains several rectangles, each representing a *section* of the total range.

The **Map** column is divided into 32 sections. The amount of memory represented by each section varies, depending on the sequencer memory depth available on the system.

Section Column

The column called `Section` is displayed to the left of **Map**. Each section contains the amount of memory represented by a single rectangle in the **Map** column.

The `Section` column is also divided into rectangles, each rectangle representing one *page* of the Error Map. The number of pages in the `Section` column varies, depending on how much memory you have in your tester.

One page of Error Map data always holds 1k of data.

The following table show the amount of memory represented by the column `Section` for various sequencer memory sizes.

Table 8-1. Error Map Section Sizes

Model	Sequencer Memory	Section Size
D50	32k	1k (1 page)
	128k	4k (4 pages)
	512k	8k (8 pages)
D100, D200, and D400	64k	2k (2 pages)
	256k	8k (8 pages)
	1M	16k (16 pages)

Note 👆 In a tester with a sequencer memory of 32k, the size of a section is the same as one page of Error Map data (1K). In this case, the Section column is not displayed.

Error Map Page   The error map itself is the grid of dots in the center of the window. The results of your test are displayed, 1k at a time, in this grid. You can scroll through the pages of the error map by:

- using the scroll bar on the right of the window;
- entering the cycle number or vector number in the entry fields on the left of the window;
- clicking a rectangle in the Map or Section column;
- or by clicking (Next Error) or (Prev Error) to find the next failed test cycle.

Passed test cycles are indicated by a small dot in the corresponding position in the grid. Failed test cycles are indicated by a diamond.

The cycle number is listed down the left hand column of the page.

Interpreting Test Results

As an example, consider the following situation: a functional test generates 24k cycles of receive data in a D200 tester with 64k of receive data memory. In this tester configuration, each section consists of 2k of data. The Map column, therefore, displays 12 rectangles.

If, for example, your test requires a vector range of 64k, all 32 rectangles would be displayed. If one or more errors are detected on a page, the rectangle representing that page comes up black.

An example of an error map for a test with 48k cycles is shown in Figure 8-2. The tester has a sequencer memory of 64k.
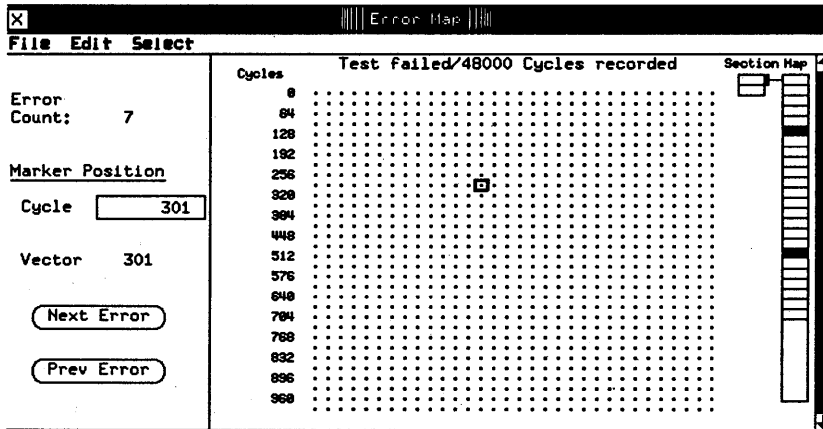
Figure 8-2. Error Map Window - Global Overview of Errors

At the top of the display area in Figure 8-2 a message informs you:

`Test failed/48000 Cycles recorded`

To find out how many errors have occurred, look at the Error Count in the left hand area of the Error Map Window. You can see that seven errors have occurred. By looking at the **Map** column, you can see two black rectangles. The seven errors have occurred somewhere on those two pages.

Note   The `Error Count` corresponds to the number of *recorded* cycles. If your test produces more cycles than fit into the sequencer memory, only the errors stored in the *last* sequencer memory block are counted and the count displayed at the end of the test.

Test Failed Flag   The moment an error is detected by a test performed in
the RTC Mode, the Test Failed Flag is set. The messag
at the top of the Error Map Window is derived from thi
flag. The same goes for the Test Passed/Failed message
that appears at the end of each test in the Report
window.

You may see the Test Failed Flag set although there are
no errors to be seen on the Error Map. Alternatively,
there may be errors visible, but the flag is cleared. In al
there are four main possibilities:

- Test Failed Flag Set with Errors Displayed
- Test Failed Flag Set with No Errors Displayed
- Test Failed Flag Cleared but Errors Detected
- Test Failed Flag Cleared but No Errors Detected

Test Failed Flag Set - Errors Displayed

This is the normal fail condition. Errors have been
detected and are visible on the Error Map.

Test Failed Flag Set - No Errors Displayed

This is a fail condition - an error somewhere in the
test has caused the flag to set. However, more cycles
have been received than fit in the memory and the
early cycles where the error has occurred have been
overwritten by subsequent error free cycles. The error
(or errors) are therefore not visible in the last cycles.

Test Failed Flag Cleared - Errors Detected

This is an artificial condition, but one rather useful in
sequencer programming. You may want to program the
sequencer to take some sort of action on detecting an
error, for instance a jump to some sequencer instruction.
Having taken the jump, you reset the Test Failed Flag,
ready for the next error.

You can enable or force to clear the Test Failed Flag in the RS column of the Sequencer Programming window.

Therefore, this condition is a fail condition, because errors have been detected.

Test Failed Flag Cleared - No Errors Detected

This can mean one of two conditions:

- The normal pass condition. No errors have been detected on any of the cycles and therefore no errors are visible.

- A fail condition that has been forced to pass. You have programmed one of the sequencer instructions in the Sequencer Programming window to reset the Test Failed Flag. Any errors detected have occurred in the early part of the range and have been overwritten by subsequent cycles.

In order to make sense of the results, you must keep track of the status of the Test Failed Flag.

Checking for Errors in a Section of Data

To get to a section where errors have been recorded, click the relevant black rectangle in Map. This situation is shown in Figure 8-3.

There are still no individual errors to be seen on the current page although the correct section has been selected.

Passed cycles are displayed as small dots, failed cycles are displayed as large diamond-shaped dots.

One or more errors detected on a page causes the rectangle representing that page in Section to come up black.
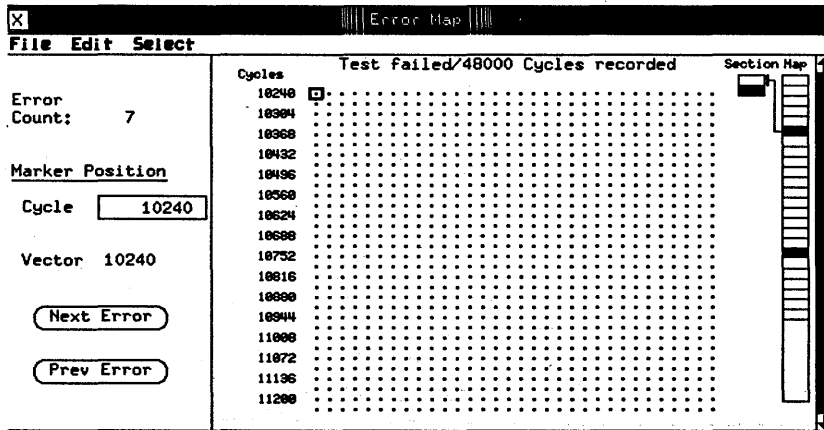
Figure 8-3. Error Map Window - Accessing a Section of Data

The currently selected page is referenced to the section it is part of by a connecting line running between the column Section and the column Map. Figure 8-3 shows the first page is selected.

The errors are on the second page. To see them you hav to click the black rectangle, which represents the second page of error data. This situation is shown in Figure 8-4

Checking for Errors on a Page of Data

Notice that the connecting line now references the second page. You can now see a single error at vector 12034. This could mean just one faulty channel or a number of faulty channels in this cycle, or in any of the other cycles that have failed in this test.
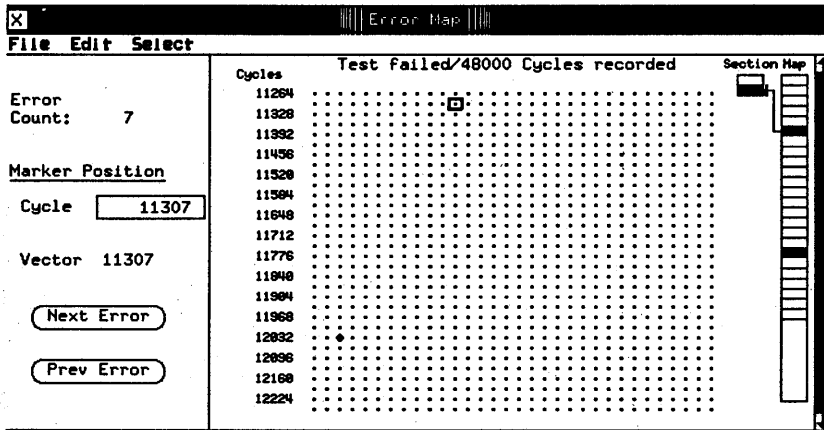
Figure 8-4. Error Map Window - Accessing a Page of Data

For every cycle the errors from all configured machine channels are ORed together and the result is displayed either as a small dot for a pass cycle or a large dot for a fail cycle. To find out which pin or pins have failed in a particular cycle, refer to the section "Reading the Pin Error List ."

Figure 8-5 shows the rest of the errors in this test. They are at cycles 35782 to 35787. Notice that the connecting line between Section and Map now references the first page in Section to the second black rectangle in Map. Notice also that the scroll box has moved next to the section that is being accessed.
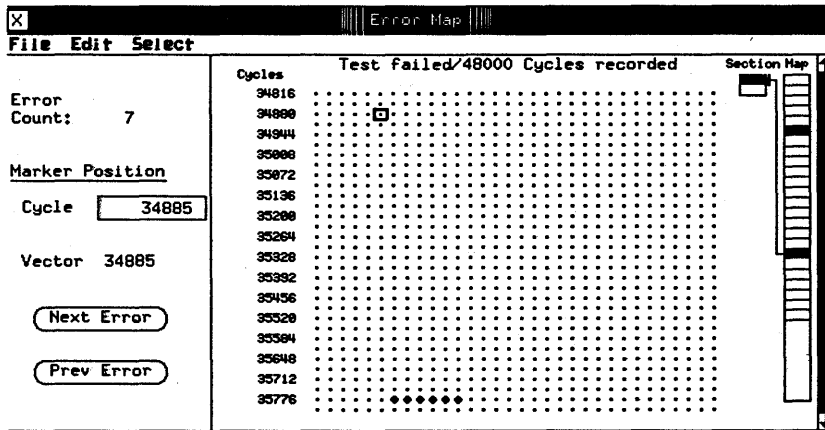
Figure 8-5. Error Map Window - Directly Accessing a Page of Error Data

Scrolling the Error Map

The alternative to selecting sections in the Map column i the scroll box. You can scroll a page of Error Map data with a resolution of a single row. As you scroll the Error Map the reference line connecting Section and Map also moves over the Section column to notify you of your approximate position within a page.

Searching for Errors

A quick way of locating errors is to click the (Next Error) or (Previous Error) pushbuttons. To get for instance to the second error (shown in Figure 8-5) you only need to clicl (Next Error) twice.

This hypothetical test has produced only seven errors. If a test comes up with hundreds of errors it is easier to scroll to the area of interest and then start looking at individual errors.

Cursor Positioning

The Error Map cursor has the shape of a small box. Its current position is given in terms of a `cycle` and a `vector` number in the left area of the Error Map Window.

Note

When no data for the Error Map is available, the cursor, and the cycle and vector numbers are not displayed.

You can move the cursor by clicking the mouse on any part of the error map where cycles are displayed, or as already discussed, by clicking on (Next Error) or (Previous Error) pushbuttons. The cycle number always reflects the position of the cursor.

Alternatively, you can type a number into the `Cycle` entry field or move the cursor with the `Cycle` vernier.

Difference between the Vector Number and the Cycle Number

In the above test the cycle and vector numbers were identical. This was because the vector sequence was linear and proceeded from vector 0 to the last vector, whereupon the test stopped.

In many tests the cycle and vector numbers will not be the same. This will depend on how you program the Vector Sequencer. The vector sequence may not start at zero but at some value in the vector range. There may be initialization vectors before the actual test vectors. There may be jumps in the vector sequence which depend on the evaluation of certain test conditions, and there may be loops where one or more vectors are looped several times before the vector sequence is allowed to continue. While this is going on, the cycle number keeps incrementing, starting at the first cycle and finishing when the test ends.

Break Cycles

Break cycles result from break vectors being downloaded to the DUT. Break vectors are downloaded to the DUT under the following conditions:

- The number of vectors programmed (per sequencer instruction) is not a multiple of four. Break vectors may be generated at the beginning of the first sequencer instruction and when the sequencer changes state.

- The sequencer has been programmed to download break vectors.

When a break vector is generated the **Vector** number is replaced with the word **BREAK**. You are responsible for tracking the break vectors generated by your particular test and for knowing why they occurred.

Effects of Bit and Word Mask

The bit mask masks off individual bits in a vector. Errors occurring on the bits in the vector sequence that have been masked off are ignored. Bit masking is especially useful in the DA Mode, where the received bit can individually be inspected on the State List Page. The bit mask cannot be used in the 50, 200 and 400 modes, since both receive bits per channel are used as receive data bits.

The word mask masks off the whole vector. Any errors occurring in a masked vector are ignored. The word mask will often be used in the modes in which the bit mask is not available.

Note ☞ An error occurring in a vector which is suppressed by th word mask, will still be displayed on the Pin Error List.

Reading the Pin Error List

The Pin Error List is another representation of failures. It shows the failed pins. If one or more errors register on a pin in the whole of the vector range, the affected pin flags an error in the Pin Error List.

The Pin Error List does not show *failed pins per vector*. It shows *failed pins per test*. If you require a detailed list of failed pins in each vector you have to run the functional test in Data Acquisition mode (test function acquire data) and look at the results on the State List Window or the Timing Diagram Window.

To display the Pin Error List, click the Select option in the menu bar, then click pin error list.

Scrolling the Pin Error List

The Pin Error List shows four pins per row in 23 columns. If your device has more than 88 output pins, you will have to scroll the list to see them all. You can scroll the list with a resolution of a single row of pins.

Interpreting the Pin Mask

The pin mask is set in the Vector Setup Window under (Select) and (show special vectors) by inserting an X in the *output pin* column of the pin or pin group you wish to mask. When set, the affected pins are masked off and any errors occurring on these pins are ignored.

The names of the pins that have been masked off in the Vector Setup Window are preceded by an X in the Pin Error List. These pins never show an error.

Comparing Error Data on Two Displays

The Error Map and the Pin Error List are automatically updated at the end of each functional test unless you have disabled the update function. This is the default setting - update is on when you start up the Error Map Window. When update is off, the Error Map and the Pin Error List are not updated by a new test. This way the old error information is frozen on the screen and can be used for comparison purposes.

To disable update, click the mouse on (Select) in the main menu bar and pick (update OFF). To enable again pick (update ON). While update is disabled the mouse pointer takes the form of a cross and the Vector numbe disappears.

To make a comparison of results from two similar tests, use the following procedure:

1. Open an Error Map Window.
2. Run a functional test.
3. Disable update to freeze the current error display.
4. Open a second Error Map Window.
5. Modify your test setup.
6. Run the modified test. The error data will be dumped to the second Error Map Window.
7. Compare the two error displays.

Additional Considerations When Disabling the Update Facility

If you change the pin configuration or the Vector Setup or run a test in the DA Mode while the update is disabled, the result area clears on re-enabling update. This prevents the display of result data that is no longei compatible with the current setup.

If you change the pin configuration or the Vector Setup or run a test in the DA Mode while the update is enabled, the vector number disappears. This is because the relationship between vectors and cycles is no longer the same.

Caution

If you have two Error Map Windows running, make sure one has the update disabled. Failure to observe this will result in only one of the Pin Error Lists being updated, however you will not be certain which one. This caution is relevant only to the Pin Error List, not to the Error Map itself.

Obtaining a Hard Copy of Error Map Window

To dump the Error Map or the Error Pin List to the printer click (File) in the menu bar and select (list).

Referencing a Cycle with the Global Buffer

The Cycle entry field not only shows the position of the cursor on the Cycle Error Map. You can also type a number in there to look at a page of error data around the entered cycle.

The Cycle entry field is part of the global buffer used in all setup and result windows. Using this buffer you can transfer all types of information between various windows. The use of this buffer in the Error map Window is useful when referencing a cycle in the State List or Timing Diagram Window.

Example: You have run a functional test in the RTC Mode and received unexpected errors in a part of the vector sequence. You wish to make a more detailed examination of this part of the vector sequence by displaying the errors in the State List Window.

You copy the desired cycle into the buffer (by using (Edit) and (copy)) and run the functional test in the DA Mode by executing the test function acquire data. You then call up the State List Window and display the referenced cycle (by clicking (Edit) and (paste)).

9

Using the State List

What the State _ist Does

The State List window enables you to see the results of functional tests performed in the Data Acquisition (DA) Mode. The other window displaying results in the Data Acquisition Mode is the Timing Diagram. Results of functional tests carried out in the Real Time Compare Mode appear in the Error Map Window.

The State List Window uses a vertical, list format to display the drive and receive data for the test just executed. It provides failure information down to individual bits.

In the Data Acquisition (DA) Mode receive data is stored in the receiver pattern memory. Error checking (if enabled) takes place in the Programming Workstation (PWS), where received and expected data is compared.

The Test Function, acquire_data performs a functional test of the DUT and outputs the result of the test to the State List Window. You can also use the Test Function start_sequencer with the DA Mode selected.

You would normally perform a functional test in the RTC Mode, at first (with the test function functional). You would set the test parameters to relaxed values. If errors occur, you would repeat the functional test in the DA Mode at the same relaxed values (using the test function acquire_data). From the results of this test, you can see precisely which data bits have failed.

Activating the State List Window

To activate the State List, click the key (State List) in the Main Menu. The State List window appears in the lowei left hand corner of the screen.

As Figure 9-1 shows, the window is divided horizontally and vertically into areas. Taking the horizontal division first, the pin and group names are in the top half of the window, the corresponding data is in the bottom half. The vertical division creates a left hand area which gives the current cycle and vector numbers and a right hand area which shows the result of the functional test.


```
X                              ||| State List ||||
File  Edit  Select  Format  Help
Pin or Group    C D D D D N S S N Q Q Q Q A A D M Q
Names           l 3 2 1 0 C 2 1 C 3 2 1 0 1 1 i o o
                o     i   1 o     1 1 n d u
                c     n   u     i o   e t
                k         t     n u

Cycle  Vector   I I I I I I I I 0 0 0 0 0 I 0 I I 0

    0     0    1 0 0 0 0 0 0 0 0 0 0 0 0 80 00 0 0 0
    1     1    1 0 0 0 0 0 0 1 0 1 0 0 0 1 82 11 0 2 1
    2     2    1 0 0 0 0 0 0 1 0 1 0 0 1 0 82 12 0 2 2
    3     3    1 0 0 0 0 0 0 1 0 1 0 0 1 1 82 13 0 2 3
    4     4    1 0 0 0 0 0 0 1 0 1 0 1 0 0 82 14 0 2 4
    5     5    1 0 0 0 0 0 0 1 0 1 0 1 0 1 82 15 0 2 5
    6     6    1 0 0 0 0 0 0 1 0 1 0 1 1 0 82 16 0 2 6
    7     7    1 0 0 0 0 0 0 1 0 1 0 1 1 1 82 17 0 2 7
    8     8    1 0 0 0 0 0 0 1 0 1 1 0 0 0 82 18 0 2 8
    9     9    1 0 0 0 0 0 0 1 0 1 1 0 0 1 82 19 0 2 9
   10    10    1 0 0 0 0 0 0 1 0 1 1 0 1 0 82 1A 0 2 A
   11    11    1 0 0 0 0 0 0 1 0 1 1 0 1 1 82 1B 0 2 B
```

Figure 9-1. State List Window

Viewing Results in the State List Window

The State List has the usual scroll bars to allow you to change the cycles being displayed in the window. As with the Vector Setup window, you can also click any number in the Vector or Cycle column, and type in the vector or cycle number that you want to view.

Increasing the Window Size

You can increase the size of the display, to show more cycles at a time, by clicking the `double size` option in the `Select` pulldown menu.

Data Representation in the State List Window

The State List shows both drive data and receive data. All pins defined in the Pin Configuration Window can be displayed. The pins or pin groups currently being displayed will have been defined in the `Display table format` dialog box.

Differences between State List and Vector Setup displays

The differences lie in data orientation and the type of data displayed.

The Vector Setup is vector oriented. Each line of data in the Vector Setup represents a test vector. A test vector consists of drive data and expected data.

The State List is cycle oriented. Each line of data represents received data from the DUT as well as corresponding drive data for that cycle. An example of a State List Window with data present is given in Figure 9-2.

Figure 9-2. State List Window - Data Present

In the process of a test a drive data vector may be output to the DUT once or several times. Each time a drive data vector is output, the tester is said to have executed a *cycle*.

As tests often make use of vector jumps or loops there may be many more cycles than vectors. The State List shows all cycles that have been executed in a test, starting at cycle 0 and finishing at cycle n-1, where n is the actual number of cycles recorded.

Positive and Negative Cycles

The way the State List displays data is closely related to the settings made in the Vector Sequencer Control and the Vector Sequencer Programming screens of the Vector Setup Window. The Delay Counter, set in the Sequencer Programming Screen of the Vector Setup Window, affects the cycle numbering in the State List Window. Cycle number 0 is where the Delay Counter starts counting. All vectors prior to this point produce initialization cycles, which are shown negative. All vectors after this point produce test cycles, which are

shown positive. A full description of the Sequencer
Programming Screen is given in *Advanced Testing with
the HP 82000*.

When interpreting the State List display there are the
following basic points to note:

- The Vector column always shows the vectors that
  have been sent to the DUT, regardless of whether they
  are part of a linear vector sequence or contain loops
  or jumps. The Cycle column shows the cycles which
  result from the executed vectors.

- Any initialization vectors (between the
  initialization start vector and the first test
  vector) result in negative cycles. The first test
  vector results in Cycle 0. Subsequent test vectors
  result in positive cycles.

- If initialization as well as test vectors are sent to
  the DUT, the Delay Counter, when enabled, starts
  counting down at the first test vector. It can be
  used to set the length of the test, or to display a
  selected portion of the received data cycles. The
  Delay Counter is accessible via the Vector Sequencer
  Programming screen. Detailed description of the Delay
  Counter and its uses are given in *Advanced Testing
  with the HP 82000*.

- When the Delay Counter has been disabled the test
  produces only negative cycles.

- Looping a series of vectors results in the looped vector
  range being repeated in the Vector column, while the
  corresponding cycles are shown as a linear range in the
  Cycle column.

Break Cycles

Figure 9-3 shows a State List Window with break cycles detected. Break cycles are notified with BREAK in the Vector column.

Break cycles occur during the execution of break vectors or when the break state has been programmed.

```
[X]                                 ||| State List |||
 File  Edit  Select  Format  Help
 Pin or Group          |C|D|D|D|D|N|S|S|N|Q|Q|Q|Q                    [↑]
 Names                 |1|3|2|1|0|C|2|1|C|3|2|1|0
                       |o|       |i| |   |o|
                       |c|       |n| |   |u|
                       |k|               |t|

    Cycle   |  Vector  |I|I|I|I|I|I|I|I|0|0|0|0|0

      -11         0     0 1 0 0 0 Z Z Z 0 0 0 0 0
      -10         1     1 0 0 0 0 Z Z Z 0 0 0 0 0
       -9         2     1 0 0 0 0 Z Z Z 0 0 0 0 0
       -8         3     1 0 0 0 0 Z Z Z 0 0 0 0 0
       -7         4     1 0 0 0 0 Z Z Z 0 0 0 0 0
       -6         5     1 0 0 0 0 Z Z Z 0 0 0 0 0
       -5         6     1 0 0 0 0 Z Z Z 0 0 0 0 0
       -4         7     1 0 0 0 0 Z Z Z 0 0 0 0 0
       -3      BREAK     0 0 1 0 0 Z Z Z 0 0 0 0 0
       -2      BREAK     0 0 1 0 0 Z Z Z 0 0 0 0 0
       -1      BREAK     0 0 1 0 0 Z Z Z 0 0 0 0 0
        0        11     1 0 0 0 0 Z Z Z 0 0 0 0 0
[←]                                                               [→]
```

Figure 9-3. State List Window - Break Cycles

Obtaining a Hard Copy of State List Window

To dump the State List to the printer click File in the menu bar and select hardcopy.

Printing Out the State List

From the File menu, the print option allows you to print-out all, or part, of the State List. Figure 9-4 show the print dialog box.

```
┌─────────────────────────────────────────────┐
│                                               │
│   ● print all                                 │
│   ○ print all, show errors                    │
│   ○ print errors only                         │
│                                               │
│      ☐ abort after  │    1051   │ errors max. │
│                                               │
│                                               │
│                                               │
│                                               │
│   ● all lines                                 │
│   ○ line range  │        0     │  │   1050   ││
│                                               │
│                                               │
│                                               │
│                   ( print )   ( cancel )      │
│                                               │
└─────────────────────────────────────────────┘
```

Figure 9-4. Print option dialog box

If you select a print which shows errors, the print-out
uses the following notation to indicate failed values:

.	indicates a failed '0', (i.e. receive data = 0, expected data = 1 or i)
/	indicates a failed '1', (i.e. receive data = 1, expected data = 0 or i)
.	indicates a failed 'i', (i.e. receive data = i, expected data = 1 or 0)
#	indicates a failed digit, when using decimal or hex notation

The **abort after** checkbox allows you to limit the print
to a maximum number of errors. The **line range** option
allows you to select which area of the State List is to be
printed.

Printing to a File

The print to file option in the Select menu allows you to send the print output to an HP-UX file. The dialog box has an additional entry field for the filename.

Referencing a Cycle with the Global Buffer

Data in any entry field in the Cycle column can be copied to or from the global buffer. This buffer is used in all setup and result windows. You use it to transfer all types of information between the display windows. In the State List Window the buffer is useful when referencing a cycle in the Timing Diagram or the Error Map Window and vice versa.

To make a transfer, you can use the copy and paste functions in the Edit pulldown menu, in the usual way. An example of transferring information between the Error Map and the State List Windows is described in Chapter 8.

Comparing Error Data on Two Displays

The State List is automatically updated at the end of each functional test run in the DA Mode unless you have disabled the update function. This is the default setting - update is on when you start up the State List Window When update is off, the State List is not updated by a new test. This way the old data is frozen on the screen and can be used for comparison purposes.

To disable update, click the mouse on Select in the main menu bar and pick update OFF To enable again pick update ON. While update is disabled the features of the State List Window change as follows:

- The mouse pointer takes the form of a cross.
- The Vector number disappears.
- The scroll bars fill up. You cannot scroll the display area.

To make a comparison of results from two similar tests, use the following procedure:

1. Open a State List Window.
2. Run a functional test in the DA Mode.
3. Scroll the display to the area of interest.
4. Disable update to freeze the current State List display.
5. Open a second State List Window.
6. Modify your test setup.
7. Run the modified test. The new data will be dumped to the second State List Window.
8. Scroll the display to the area of interest.
9. Compare the two data displays.

If you change the Pin Configuration, the Vector or Timing Setup, or run a test in the RTC Mode while update is disabled, the **Vector** number disappears and the State List remains frozen after re-enabling update. This prevents the display showing result data that is no longer compatible with the current setup. You have to run a functional test in the DA Mode to update the window.

Effects of Bit, Word Mask and Pin Mask

The bit mask masks off individual bits in a vector. Errors occurring on the bits in the vector sequence that have been masked off are ignored. The bit mask is set in the Vector Setup Window and is discussed in Chapter 6.

The bit mask cannot be used in the 50, 200 and 400 Modes since both receive bits per channel are used as receive data bits.

The word mask masks off the whole vector. Any errors occurring in a masked vector are ignored. The word mask will often be used in the 50 mode, 200 mode, and 400 mode, since the bit mask does not function in these modes. The word mask is set in the Vector Setup Window and is discussed in Chapter 6. In order to use word mask you have to set up a dedicated channel for word mask use. This is discussed in Chapter 3.

The pin mask masks off all data on a pin. When set, any errors occurring on the masked pins are ignored. The pin mask is set in the Vector Setup Window and is discussed in Chapter 6.

Checking for Errors

The State List always shows the drive and receive data. In addition, it allows you to display errors down to individual bits. A failed bit means an error has occurred in a particular vector on a particular channel.

The error display is off by default. You can enable the error display from the Select menu, by clicking ALL Cycles, Show Errors ON. To disable it, click ALL Cycles, Show Errors OFF. Any errors are displayed in inverse video, so that you can see at a glance which pins have failed on which cycle.

When Show Errors ON is selected, you can quickly locate failed cycles by using the (Next) and (Previous) keys to scroll the display from to the next or previous error.

In addition to this, two error search options in the Select menu allow you to list only those cycles which have failed:

- Errors ONLY, Search from TOP - searches from the first cycle in the State List, and consecutively lists all the cycles which have errors.
- Errors ONLY, Search from ACTUAL POSITION - searches from the first cycle currently displayed, and consecutively lists all the cycles which have errors.

The State List gives no information about what the correct data should have been, and it is not always possible to deduce it. For instance if you have defined pin groups for display (via the Define table format - see below in "Selecting Pins for Display"), you will see a hex number in inverse video, and will not be able to work out the correct data at all. To see the correct data

look at the expected data vectors displayed in the Vector Setup Window.

Checking for Glitches

In some modes *Dual Level Compare* in conjunction with *Window Strobe.* In Dual Level Compare the tester checks the DUT output signals against two compare thresholds. In Window Strobe the tester uses two timing edges to create a time window in which it looks at the DUT output signals. This combination enables you to check for the presence of glitches and intermediate states.

A glitch occurs when the signal received from the DUT crosses at least one of the compare thresholds one or more times inside the time window. A glitch is displayed in the State List Window as a U see Figure 9-5. The question marks (?) are displayed instead of numbers in hexadecimal notation and signify groups which cannot be reconstructed since one or more of the pins have come up as glitches. Where more than one digit appears in a group column, the group contains more than four channels.

```
[X]                          |||| State List ||||
File  Edit  Select  Format  Help
Pin or Group    C D D D N S S N Q Q Q Q A  A  D  M Q
Names           1 3 2 1 0 C 2 1 C 3 2 1 0 1  1  i  o o
                o       i   o         1  1  n  d u
                c       n   u         _  _  d  e t
                k           t         i  o  e
                                      n  u

 Cycle   Vector I I I I I I I I 0 0 0 0 0 I  0  I  I 0

      0      0  1 0 0 0 0 0 0 0 0 i i U i 80 0? 0  0 ?
      1      1  1 0 0 0 0 0 1 0 1 i i i U 82 1? 0  2 ?
      2      2  1 0 0 0 0 0 1 0 1 i i U U 82 1? 0  2 ?
      3      3  1 0 0 0 0 0 1 0 1 i i 1 U 82 1? 0  2 ?
      4      4  1 0 0 0 0 0 1 0 1 i U U U 82 1? 0  2 ?
      5      5  1 0 0 0 0 0 1 0 1 i i i U 82 1? 0  2 ?
      6      6  1 0 0 0 0 0 1 0 1 i i U U 82 1? 0  2 ?
      7      7  1 0 0 0 0 0 1 0 1 i i 1 U 82 1? 0  2 ?
      8      8  1 0 0 0 0 0 1 0 1 U U U U 82 1U 0  2 U
      9      9  1 0 0 0 0 0 1 0 1 i i i U 82 1? 0  2 ?
     10     10  1 0 0 0 0 0 1 0 1 1 i i U 82 1? 0  2 ?
     11     11  1 0 0 0 0 0 1 0 1 i i 1 U 82 1? 0  2 ?
```

Figure 9-5. State List Window - Glitch Detection

Detecting Intermediate States

An intermediate state occurs if the signal received crosses above the low level threshold and below the high level threshold. An intermediate state is marked in the State List Window with an i, see Figure 9-6. The question marks (?) are displayed instead of numbers in hexadecimal notation and signify groups which cannot be reconstructed since one or more of the pins have come up in the intermediate state. Where more than one digit appears in a group column, the group contains more than four channels.

```
┌─────────────────────────────────────────────────────────────────────────────────┐
│ ☒                              ║║║ State List ║║║                                  │
├─────────────────────────────────────────────────────────────────────────────────┤
│ File   Edit   Select   Format   Help                                              │
│ Pin or Group      C D D D D N S S N Q Q Q Q A  A  D M Q                         ↑ │
│ Names             l 3 2 1 0 C 2 1 C 3 2 1 0 1  1  i o o                           │
│                   o         i     o         1  1  n d u                           │
│                   c         n     u         _  _    e t                           │
│                   k               t         i  o                                  │
│                                             n  u                                  │
├─────────────────────────────────────────────────────────────────────────────────┤
│ Cycle     Vector  I I I I I I I I 0 0 0 0 0 I  0  I I 0                            │
├─────────────────────────────────────────────────────────────────────────────────┤
│      0         0  1 0 0 0 0 0 0 0 0 i i i i 80 0i 0 0 i                            │
│      1         1  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                            │
│      2         2  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                            │
│      3         3  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                            │
│      4         4  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                            │
│      5         5  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                            │
│      6         6  1 0 0 0 0 0 1 0 1 i i 1 i 82 1? 0 2 ?                            │
│      7         7  1 0 0 0 0 0 1 0 1 i i 1 i 82 1? 0 2 ?                            │
│      8         8  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                            │
│      9         9  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                            │
│     10        10  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                            │
│     11        11  1 0 0 0 0 0 1 0 1 i i i i 82 1? 0 2 ?                          ↓ │
│ ←                                                                               → │
└─────────────────────────────────────────────────────────────────────────────────┘
```

Figure 9-6. State List Window - Detection of Intermediate State

Selecting Pins for Display

The menu bar option Format opens the Define table format dialog box on the screen. It allows you to select for display the pins you wish to see in the right hand area (the result area) of the State List Window. The dialog box operates in the same way as that used in the Vector Setup Window.

10

Using the Timing Diagram

What the Timing Diagram Does

The Timing Diagram window enables you to see the results of functional tests performed in the Data Acquisition (DA) Mode. The other window displaying results in the Data Acquisition Mode is the State List. Results of functional tests carried out in the Real Time Compare Mode appear in the Error Map Window.

The Timing Diagram window uses a horizontal, graphical format to display the drive and receive data for the test just executed. It provides failure information down to individual bits.

In the Data Acquisition (DA) Mode receive data is stored in the receiver pattern memory. Error checking (if enabled) takes place in the Programming Workstation (PWS), where received and expected data is compared.

Activating the Timing Diagram Window

To activate the Timing Diagram click the key (Timing Diagram) in the Main Menu. The Timing Diagram Window appears in the lower left hand corner of the screen.
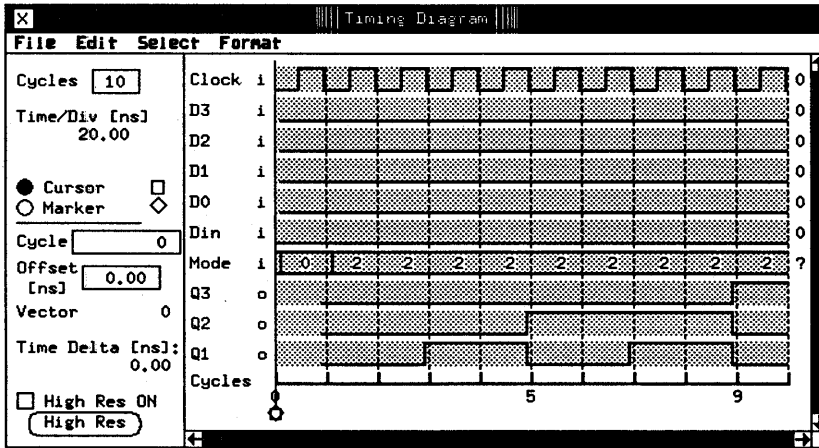
Figure 10-1. Timing Diagram Window

As Figure 10-1 shows, the window is divided into two areas, a left hand area which gives the current cycle and vector numbers, as well as other related information, and a right hand area - the results area, which displays the result of the functional test in a graphical form.

Data Representation in the Timing Diagram Window

Like the State List, the Timing Diagram can show both drive and receive data. Because the State List and the Timing Diagram show results of identical tests, the data extracted from the tester hardware for these two windows, is also identical. The differences lie only in the way the data is represented. As its name suggests, the Timing Diagram Window displays the data in the form of a timing diagram.

Like the State List, the Timing Diagram is cycle oriented. Each cycle is represented in the results area horizontally, a data cycle represents received data from the DUT as well as corresponding drive data for that cycle.

Results Area Marking

Refer back to Figure 10-1. Around the edges of the results area you see various markings and labels. The single letters in a column just to the left of the results area designate the pin type. All pins are split into their input and output components and displayed separately. This applies also to any pin groups.

i designates an input pin or pin group - drive data channel(s)

o designates an output pin or pin group - receive data channel(s)

Further to the left is a column listing the names of the pins and pin groups for which data is currently displayed. Selection of pins for display and the order in which they appear is determined in the Define table format dialog box discussed later on in this chapter under "Selecting Pins for Display." At the bottom of the column is the word Cycles. This refers to the cycle numbering along the bottom of the display area.

The column of characters to the right of the results area gives the value of the data at the cursor. This is discussed in detail in "Use of the Cursor and Marker."

Labeling of Data within the Result Area

Signals on individual pins or pin groups are displayed in a manner used by standard data sheets for digital devices. You can see the state of single pins from the waveform, that is, whether the level is low, high or intermediate. The signal values of pin groups are represented by a hex number in each cycle. Apart from this a bus cycle can be labeled with the following characters:

? represents an unknown value and occurs only in pin groups. When one or more pins in the affected cycle return an intermediate value

(between low and high) the total value for the
bus cycle cannot be calculated.

s   represents a scan cycle and occurs only on the
pin designated as the serializing input pin . A
full description of scan cycles is given in the
chapter Scan Path Testing of *Advanced Testing
with the HP 82000.*

blank   a blank in the bus cycle is caused by the
following:

■ No receive data available for the early part
of the cycle. This occurs only in the first
cycle and can be rectified by moving the first
compare edge forward.
■ Creating groups from pins with inconsistent
timing or format.

Scrolling the Timing Diagram

If your test includes more pins and produces more cycles
than fit on the result display, you will have to scroll the
display.

To see more cycles scroll the display horizontally using
the horizontal scroll bar. To see more pins scroll the
display vertically using the vertical scroll bar. The scroll
box inside each scroll bar shows you where you are
within the display. The width of the scroll box shows
you how much of the total data is currently displayed.
Figure 10-2 shows the results area scrolled to a different
part of the data stream. Notice how the position of the
scroll box in both the horizontal and vertical scroll bar
has changed.
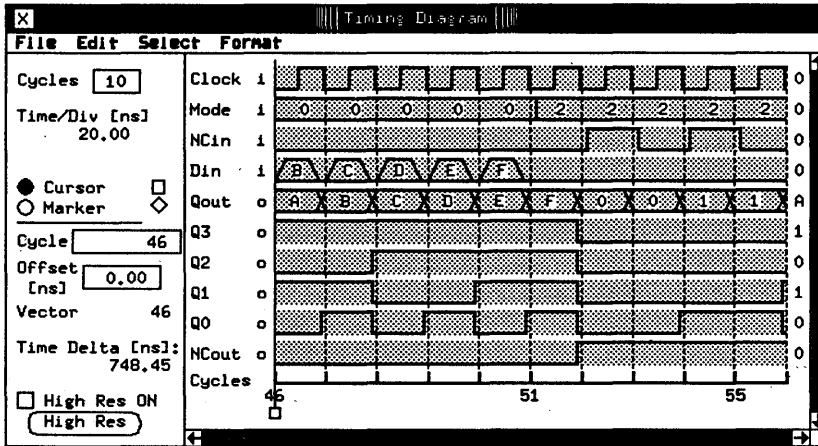
Figure 10-2. Timing Diagram Window - Using the Scroll Bars

To scroll the display, click the relevant scroll box and drag it along the scroll bar.

Horizontal Zoom

In the left hand area of the Timing Diagram Window there are a number of entry fields and other items. The first of these from the top is the entry field `Cycles`.

By clicking on this field and selecting from the fill menu, or by typing a value directly into it, you can horizontally zoom the displayed waveforms.

Note

When typing in a value, the system accepts only the values listed in the fill menu.

The Timing Diagram always displays the number of cycles shown in the `Cycles` entry field.

The right hand area of the Timing Diagram Window is marked off into ten divisions regardless of how many cycles are being displayed at a time. The value listed

under `Time/Div [ns]` tells you the time corresponding to each division in nanoseconds.

Figure 10-3 shows the zoom setting changed to 100 cycles. Notice that `Cycles` reads 100.
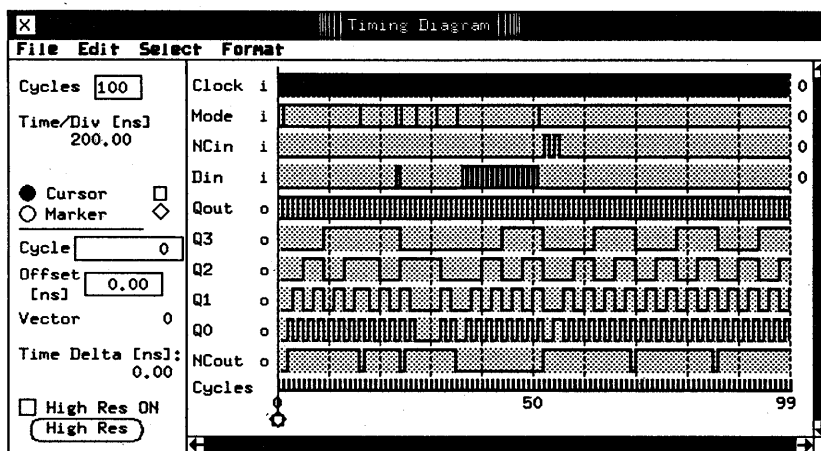
Figure 10-3. Timing Diagram Window - Using Horizontal Zoom

Use of the Cursor and Marker

The Timing Diagram Window provides a cursor and a marker for making timing measurements. These function like the timing markers on oscilloscopes or logic analyzers. Using the "radio button" you can select either the `Cursor` or the `Marker`. A "radio button" is a small circle to the left of the word `Cursor` and `Marker`. When an item is selected its radio button is filled.

Cursor

The cursor is referenced with a small square (see on the right of the word `Cursor`). When `Cursor` is selected, its position in the data stream is referenced by the readouts for `Cycle`, `Offset [ns]` and `Vector`.

The cursor can never be scrolled off the display area. When horizontally scrolling the data stream the cursor

always remains in the same place relative to the display area. There are two exceptions to this:

- you have forced the cursor to an extreme position at the beginning or end of the data stream
- the horizontal movement is less than the number of cycles currently displayed

Moving the Cursor. To move the cursor make sure its radio button is selected, then click the mouse where you want the cursor to go to. The cursor jumps to the mouse pointer. By keeping the mouse button depressed you can drag the cursor to a different position.

When clicking the mouse inside the results area, the cursor jumps to the nearest timing edge as specified in the Timing Setup Window. When clicking underneath the results area (in between the Cycles scale and the horizontal scroll bar) the cursor remains exactly where you release the mouse button.

Reading Signal Values at Cursor. The position of the cursor determines the single character (State-List-like) readout in the column to the right of the results area. The following characters can appear in the readout.

Input pins (Drive channels)

0   programmed low level

1   programmed high level

0-F   a single hex digit representing the *four most significant bits* of a pin group. If the pin group consists of more than four pins, the less significant pins are ignored.

blank   A blank occurs in pin groups containing pins with inconsistent timing or formats. Check the group components in the Timing Setup Window.

Z   Occurs for one of the following reasons:

- The affected pin has been configured as a tristate (high-Z) pin. This is only possible in 25 mode, 100 mode or 200RED mode.

- The pin has been masked with the pin mask.

?   Occurs for one of the following reasons:

- One or more pins in the affected group has been configured as a tristate (high-Z) pin. This applies only to the four most significant pins of the group.

- The affected pin has been configured as the serializing pin for scan path testing. Refer to *Advanced Testing with the HP 82000.*

- Vector number is unknown. This could be due to several reasons, all of which are described in the *Advanced Testing with the HP 82000.*

Output pins (Receive channels)

0   Low level - the level is lower than the programmed lower threshold.

1   High level - the level is higher than the programmed lower threshold.

0-F   a single hex digit representing the *four most significant bits* of a pin group. If the pin group consists of more than four pins, the less significant pins are ignored.

i   Intermediate data received. The signal lies between the programmed high and low thresholds.

-      A dash occurs only in the Window Strobe Mode, when the cursor lies outside the time window. Refer to the chapter Figure 5-1.

U   occurs for the following reasons:

■ A glitch has been detected. Such an occurrence
is defined as an *unstable* signal. This state can
only be detected for pins which use Window
compare or Dual Level Edge compare. This
condition indicates that the received signal has
crossed both thresholds more than once inside
the defined time window. A full description
of glitch detection is given a little later in
"Checking for Glitches ."

blank A blank occurs for one of the following reasons:

■ A pin group has been set up, containing pins
with inconsistent timing or formats. Check
the group components in the Timing Setup
Window.

■ No receive data is available for the early part of
a cycle (strobe not yet occurred) This occurs
only in the first cycle and can be rectified by
moving the first compare edge forward.

Figure 10-4. Timing Diagram Window - Reading the Data at Cursor

Figure 10-4 shows some of the digital read values. Note that the data waveforms do not all start at the same time. The starting positions are determined by the strobe timing - refer to the Timing Diagram. Data present before the first strobe edge cannot be defined and is therefore shown as indeterminate.

Marker

The marker is referenced with a small diamond (see on the right of the word Marker). When Marker is selected, its position in the data stream is referenced by the readouts for Cycle, Offset [ns] and Vector.

The Marker is tied to the data stream and is therefore visible only when the part of the data stream containing Marker is currently being shown in the display area.

Moving the Marker. You move the marker the same way as the cursor. See "Moving the Cursor" above. Make sure its radio button is selected first.

Selecting a Part of the Data Stream for Display

The entry field Cycle gives the position of either the Cursor or the Marker, to within a cycle. You can also move the position of either by typing a number into the entry field.

To display a part of the data stream, select either Cursor or Marker and type the desired cycle number into the Cycle entry field.

Reading the Vector Number

The number displayed to the right of the word Vector gives the current vector number. The referenced vector is associated with the cycle currently being pointed to by the Cursor or the Marker, whichever happens to be selected.

Measuring/Specifying Offset from Beginning of Period

The position of the **Cursor** or **Marker** within a cycle is specified by the **Offset**. The offset is measured from the beginning of the period, which you have selected in the Timing Setup Window. Cycle width is always equivalent to the period.

Example:

You have selected a period of 20 ns in the Timing Setup and are working in the 100 Mode. The offset has a range of 0 to 20 ns. In Figure 10-5 the cursor is set to an offset of 15 ns.
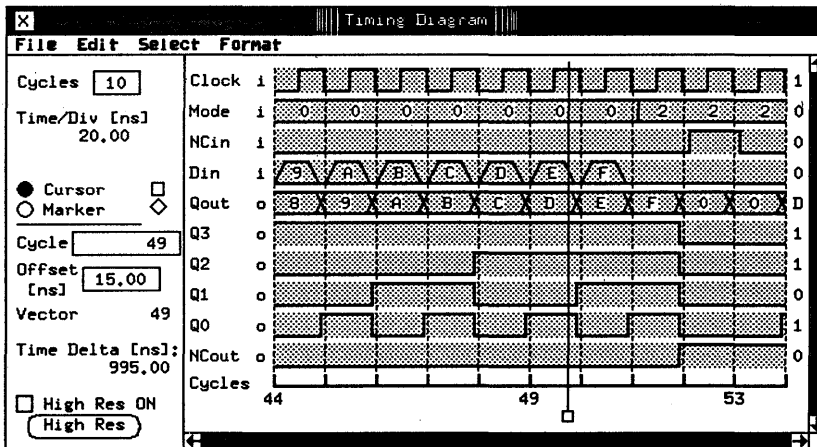
Figure 10-5. Timing Diagram Window - Measuring Offset in 100 Mode

You change the mode to 200 mode. The offset range remains at 0 to 20 ns. At the bottom of the display area the row **Cycles** now contains double the number of cycle markers. In Figure 10-6 the cursor is also set to an offset of 15 ns.
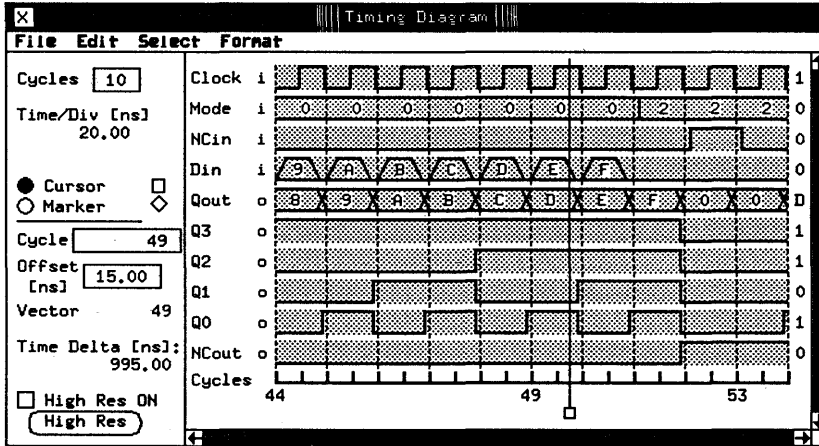
Figure 10-6. Timing Diagram Window - Measuring Offset in 200 Mode

Measuring Time Differences

Time Delta [ns] displays the relative time between the settings of Cursor and Marker. The time in nanoseconds is displayed as positive if Cursor is on the right of Marker and negative if Cursor is on the left of Marker. Figure 10-7 shows a relative time difference of 13 ns.
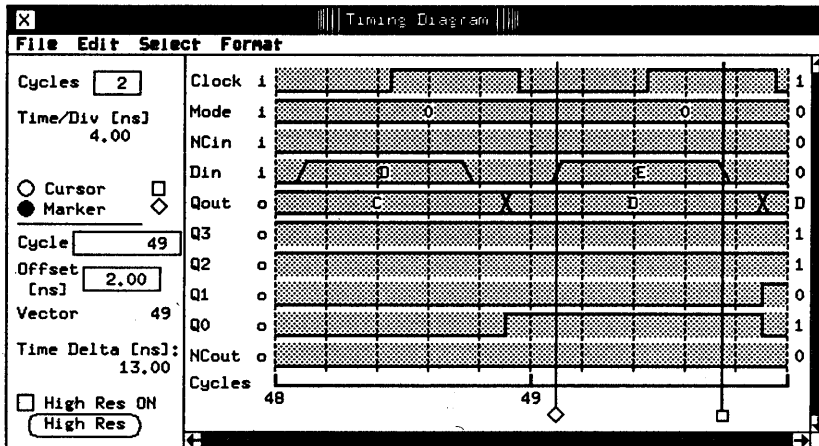
Figure 10-7. Timing Diagram Window - Measuring Delta Time

Positive and Negative Cycles

The way the Timing Diagram displays data is closely related to the settings made in the Vector Sequencer Control and the Vector Sequencer Programming screens of the Vector Setup Window. A full description of the interaction of the Timing Diagram display with the settings made in Vector Setup is given in *Advanced Testing with the HP 82000*. There are, however, a number of basic points to note when interpreting the Timing Diagram display:

- Any initialization vectors (between the initialization start vector and the first test vector) result in negative cycles. The first test vector results in Cycle 0. Subsequent test vectors result in positive cycles.

- If initialization as well as test vectors are downloaded to the DUT, the Delay Counter, when enabled, starts counting down at the first test vector. It can be used to set the length of the test, or to display a selected portion of the received data cycles. The Delay Counter is accessible via the Vector Sequencer

Programming screen. Detailed description of the Delay Counter and its uses are given in *Advanced Testing with the HP 82000*.

- When the Delay Counter has been disabled the test produces only negative cycles.

- Looping a series of vectors results in the looped vector repeated in the Vector number, while the corresponding cycles are shown in a linear fashion in the Cycle number.

Break Cycles

Figure 10-8 shows a Timing Diagram Window with break cycles detected. Break cycles are notified with BREAK in the Vector number. Break cycles occur during the execution of break vectors or when the break state has been programmed.
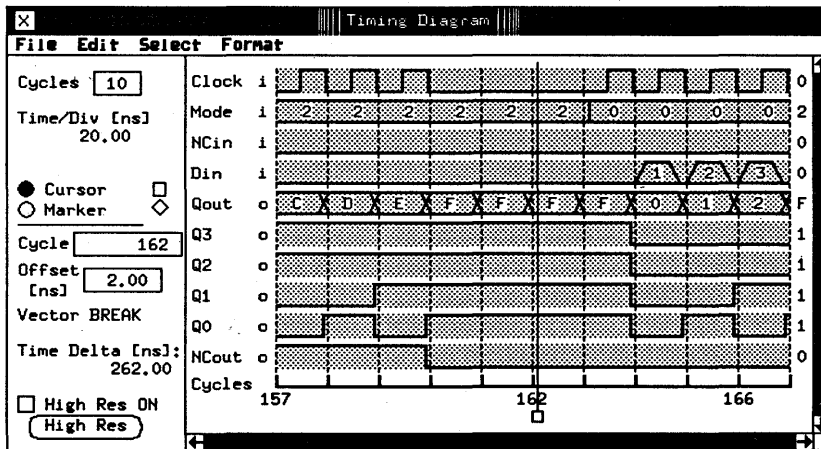
Figure 10-8. Timing Diagram Window - Break Cycles

Obtaining a Hard Copy of Timing Diagram Window

To dump the Timing Diagram to the printer click `File` in the menu bar and select `hardcopy`.

Referencing a Cycle with the Global Buffer

Data in the `Cycle` entry field can be copied to or from the global buffer. This buffer is used in all setup and result windows. You use it to transfer all types of information between the display windows. In the Timing Diagram Window the buffer is useful when referencing a cycle in the State List or the Error Map Window and vice versa.

To make a transfer you use the menu bar option `Edit` and then `copy` and `paste` as already described under "Editing Functions " in Chapter 5 in Chapter 5. An example of its use when transferring information between the Error Map and the State List Windows is described in Chapter 8.

Comparing Data on Two Displays

The Timing Diagram is automatically updated at the end of each functional test run in the DA Mode unless you have disabled the update function. This is the default setting - update is on when you start up the Timing Diagram Window. When update is off, the Timing Diagram is not updated by a new test. This way the old data is frozen on the screen and can be used for comparison purposes.

To disable update, click the mouse on `Select` in the main menu bar and pick `update OFF`. To enable again pick `update ON`. While update is disabled the features of the Timing Diagram Window change as follows:

- the mouse pointer takes the form of a cross
- the entry fields for `Cycles`, `Cycle` and `Offset` disappear. You cannot enter anything into these fields.
- The scroll bars fill up. You cannot scroll the timing display.

To make a comparison of results from two similar tests, use the following procedure:

1. Open a Timing Diagram Window.
2. Run a functional test in the DA Mode.
3. Scroll the display to the area of interest.
4. Disable update to freeze the current Timing Diagram display.
5. Open a second Timing Diagram Window.
6. Modify your test setup.
7. Run the modified test. The new data will be dumped to the second Timing Diagram Window.
8. Scroll the display to the area of interest.
9. Compare the two data displays.

Note

If you change the Pin Configuration, the Vector or Timing Setup, or run a test in the RTC Mode while update is disabled, the Timing Diagram remains frozen after re-enabling update. This prevents the display showing result data that is no longer compatible with the current setup. You have to run a functional test in the DA Mode to update the window.

Doubling the Window Size

The Timing Diagram Window can be doubled in size, thus increasing the horizontal display resolution as well as increasing the number of displayed channels from 10 to 20.

To toggle the size, click the mouse on Select and pick double size or standard size. Alternatively use the CTRL-D or CTRL-S hotkeys.

An example of the double size Timing Diagram Window is shown in Figure 10-9.
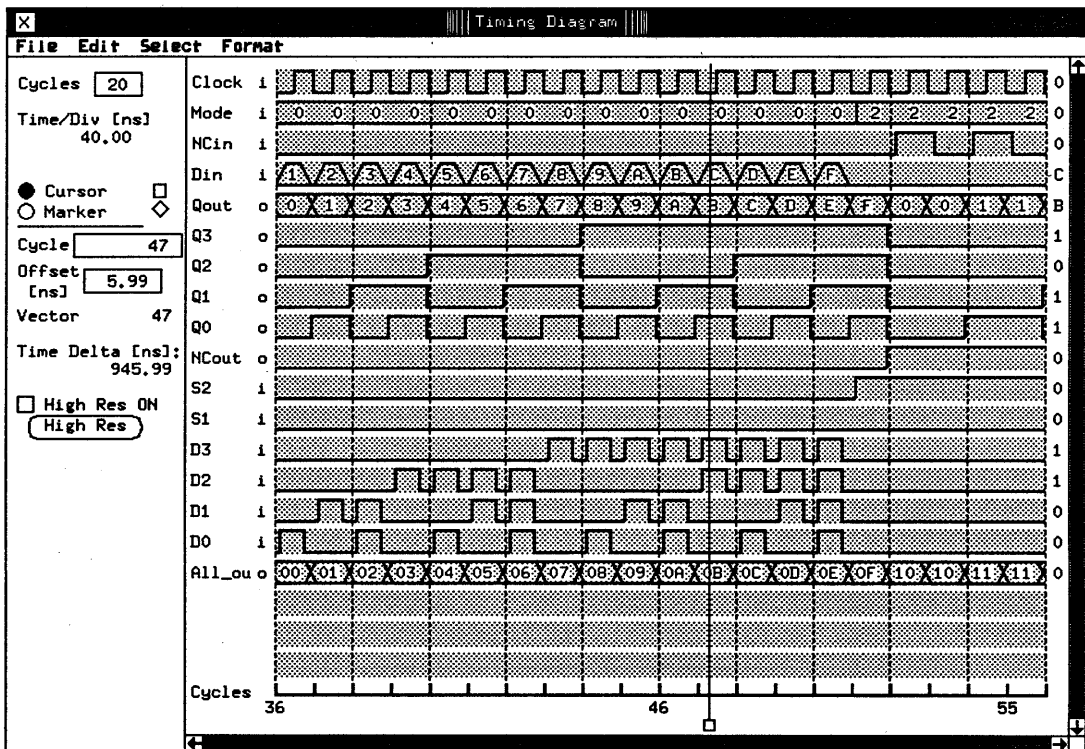
Figure 10-9. Double-sized Timing Diagram Window

Checking for Errors

The Timing Diagram always shows the drive and receive data regardless of whether the test has passed or failed. In addition, it allows you to display errors down to individual bits. A failed bit means an error has occurred in a particular vector on a particular channel.

When you enable the display of errors, what in effect you are doing is displaying the expected data. The received data is then displayed over the top of the expected data. If there are no errors the received data is superimposed over the expected data. If errors occur the expected data is visible as a separate waveform.

The error display is off by default. To enable it click the mouse on Select and pick Show Errors ON. To disable it pick Show Errors OFF. If you have a monochrome monitor:

■ errors on single channels (pins) are displayed as dashed line waveforms
■ errors in busses (pin groups) are displayed in a heavy line

If you have a color monitor:

■ errors on single channels (pins) are displayed in a different color
■ errors in busses (pin groups) are displayed with an additional line in a different color.

The color signifying erroneous data has been set to red at the factory, but you can redefine it as required. For details on programming the colors used by the HP 82000 software refer to *Advanced Testing with the HP 82000.*

An example of an error display is shown in Figure 10-10. The signal values read at the cursor are those the received data, not those of the expected data.
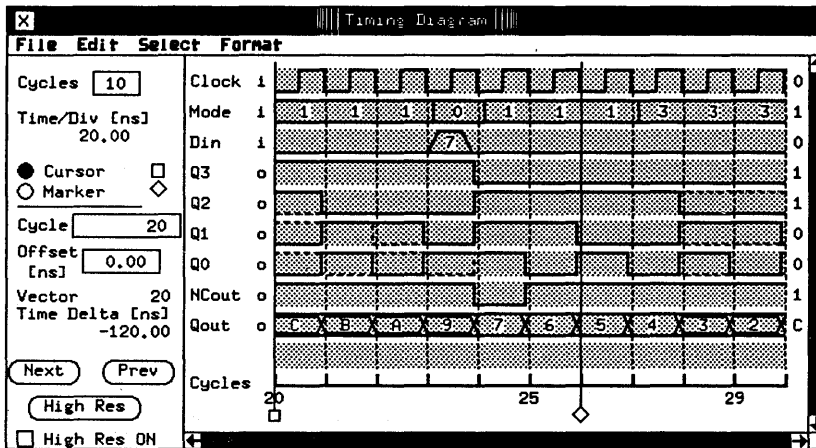
Figure 10-10. Timing Diagram Window with Error Display Enabled

When Show Errors ON is selected, you can quickly locate failed cycles by using the (Next) and (Previous) pushbuttons to move the cursor or marker to the next or previous error.

Checking for Glitches

In some pin modes the tester performs more than one comparison on each received signal. In *Dual Level Compare,* the tester checks the DUT output signals against two compare thresholds. In *Window Compare* the tester uses two timing edges to create a time window in which it looks at the DUT output signals.

These comparison methods enable you to check for the presence of glitches.

A glitch occurs when the signal received from the DUT crosses at least one of the compare thresholds one or more times inside the time window. A glitch is displayed in the Timing Diagram Window as an X see Figure 10-11.
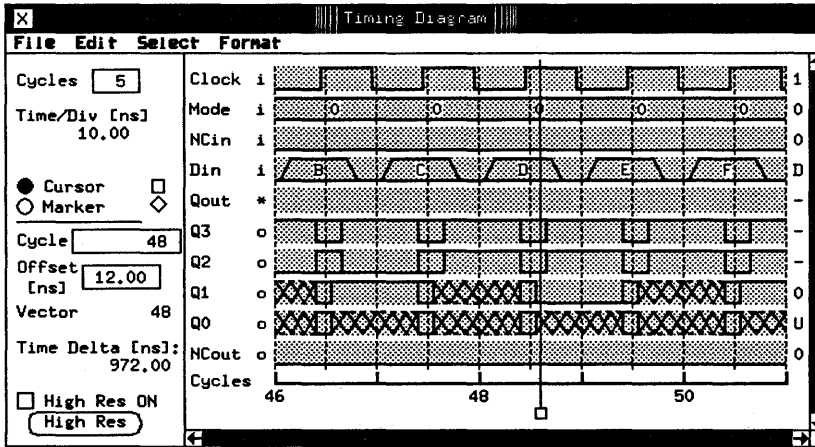
Figure 10-11. Timing Diagram Window - Glitch Detection

Selecting Pins for Display

The menu bar option `Format` opens the `Define table format` dialog box on the screen. It allows you to select for display the pins you wish to see in the right hand area (the results area) of the Timing Diagram Window. The dialog box is identical in its functionality and features to that used in the Vector Setup Window. The description of the `Define table format` will therefore not be repeated here.

Illegal Pin Groups

Pin groups defined in the Pin Configuration Window that are not allowed in the Timing Diagram Window are termed *illegal.* They are listed along with the other legal groups in the `Define table format` dialog box, but are written in the `grouplist` in shaded font and are marked in the `table format` with a star. Such groups cannot be selected for display in the results area of the Timing Diagram Window.

Example of an illegal group in the Timing Diagram is one containing both input and output pins.

High Resolution Timing Diagram

The High Resolution Timing Diagram provides a display based on high resolution timing analysis in the hardware. When you execute a high resolution test the tester does not run just one test but a whole series of tests, each time stepping the compare edge to a new position in the cycle. This way the whole cycle is sampled, resulting in more accurate information being delivered to the Timing Diagram Window.

A high resolution display shows all channels and groups that you have set up in the Define table format dialog box, the same way a standard resolution display would. However, since the data for the high resolution display is produced by multiple sampling of the DUT output pins, it is only the *single* output pins that contain high resolution timing information. Any groups of output pins that you've set up for display are also not shown in the high resolution mode.

The resolution of the video monitor and the number of cycles shown on the Timing Diagram Window play an important role in creating the display you see. The resolution of the window display will obviously be higher and *will contain more information* if you enter for instance a 2 in the Cycles entry field rather than if you enter a 50. It is important to remember that the time taken for the overall test to complete is directly proportional to the required resolution.

The number of samples per cycle the hardware makes during a high resolution test can be worked out from the following:

Standard size window	400 samples per number of cycles displayed
Double size window	600 samples per number of cycles displayed

Activating the High Resolution Timing Diagram

In the lower left hand corner of the Timing Diagram Window you see a switch High Res ON and a pushbutton High Res, see Figure 10-12. Both activate the High Resolution Timing Diagram, but work differently.
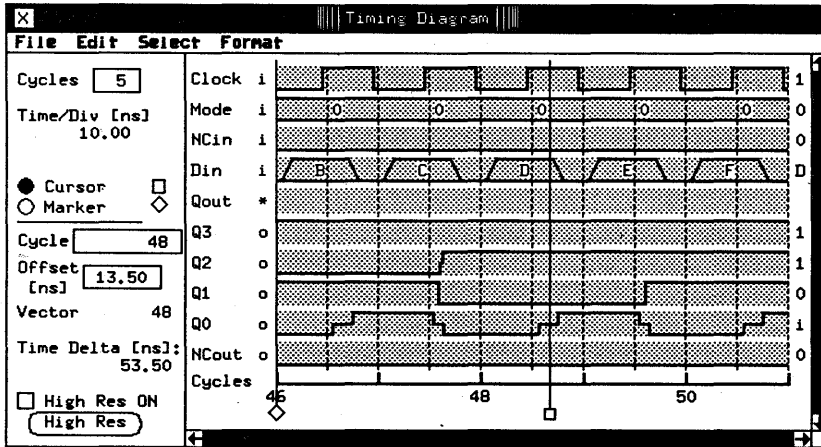
Figure 10-12. High Resolution Timing Diagram

Permanently Switching into High Resolution Mode

To select the high resolution mode click the High Res ON check box. This does two things:

1. invokes the high resolution mode
2. executes a high resolution functional test in the DA Mode

As long as the high resolution mode remains active, you can scroll the display, or alter any of the other parameters in the Timing Diagram Window, such as change the window size, move the cursor outside the displayed part of the data stream, and so on, and the output pins will remain displayed in the high resolution mode. It is important to note that any of these changes do not just redraw the display, but result in another high resolution test being performed. Every such re-test

requires a certain amount of time to run (as discussed above), and you should therefore use the high resolution mode only when you really need it.

One-off Testing in High Resolution Mode

To execute a single test in the high resolution mode click the High Res pushbutton at the lower left hand corner of the window. The system enters the high resolution mode for the duration of the test. It reverts back to standard resolution when you do any of the following:

- scroll the display
- change the window size
- change the entry in the Cycles entry field
- move the cursor outside the display part of the data stream with the Cycle or Offset [ns] entry fields
- run another test from the Test Control Window.

High Resolution Glitch Detection

If your test comes up with glitches in the standard resolution Timing Diagram you can see in which cycle they have occurred. You cannot see how many glitches have occurred per cycle and where in the cycle they have occurred. To find out you need to run the test with the High Resolution Timing Diagram enabled. Figure 10-13 shows glitches but with high resolution enabled.
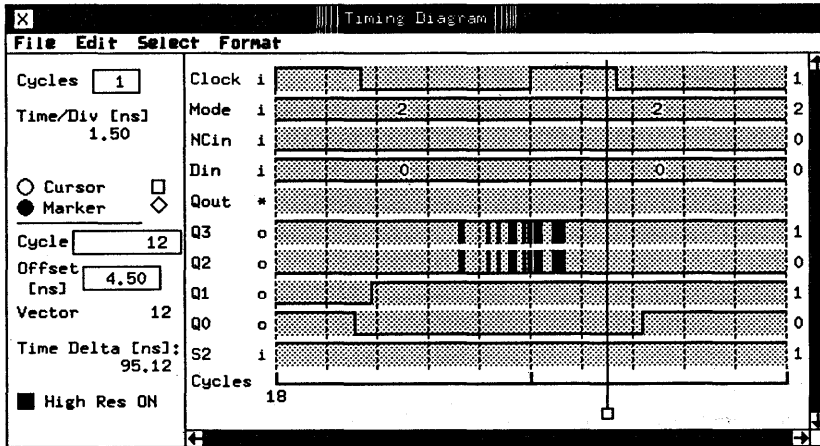
Figure 10-13. Timing Diagram Window - High Resolution Glitch Detection

As you zoom in using the Cycles entry field, the level of detail increases and you can often see glitches that were not at all visible at a lower zoom setting.

A

Test Function Error Codes

Status Messages

The following test function status will be generated at the end of test function execution.

1	Test function result is failed
0	Test Function result is passed
-1	Warning were generated during test function execution.

If errors were generated, the following codes will be generated:

MCD Errors

-2	Cannot execute firmware task successfully
-3	Internal answer buffer too small
-4	Internal task buffer too long
-5	Cannot open file
-6	Cannot read file
-7	No data in file
-8	Test file empty
-9	File is not an hp82000 file
-10	Wrong file type
-11	Wrong file revision
-12	File revision is newer
-13	file revision is older
-14	Invalid data in file
-15	Cannot write to file
-16	Cannot overwrite file

	-17	File does not exist

(

BASIC Errors

-50	Filename in Exec_tf_file too long
-51	Parameter string in Exec_tf_param too long
-52	Badly dimensioned result array
-53	Buffer problem

Test Function Process Errors

-100	TF environment failure
-101	Bad tf keyword
-102	TF parameter check failure
-103	TF execution error
-104	Configuration change during tf execution
-105	Fatal error during tf execution

Break Key

-1000	The test function was terminated with the Break key.

Automatic Timing Correction For Drive Level Variation
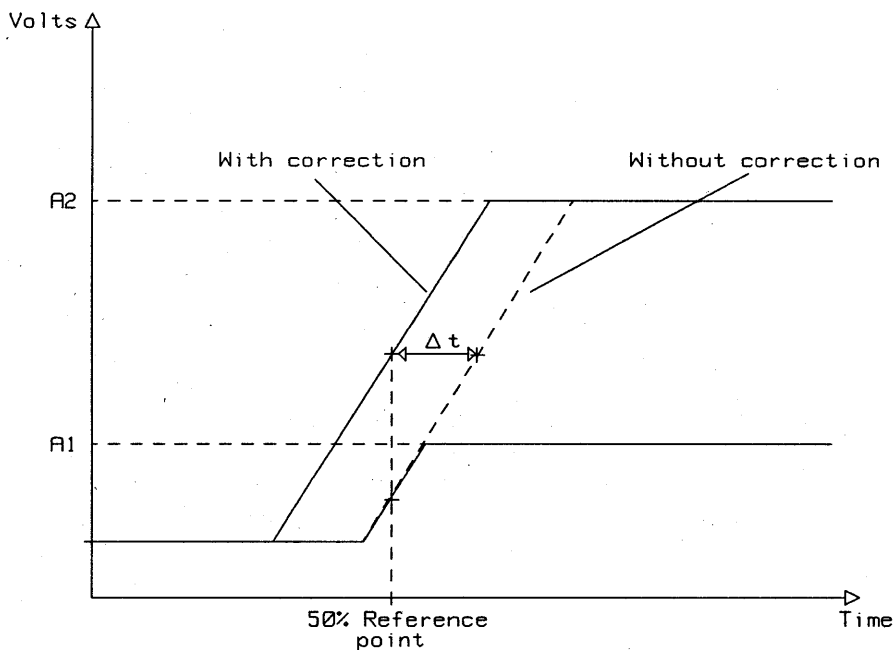
Figure B-1. Automatic Timing Correction

The HP 82000 references the signal edge timing to the instant that it reaches 50% of its transition between the low and high levels (as defined in the Level Setup window). The pin driver circuitry of the HP 82000

system exhibits a constant slew rate behavior. This causes the 50% reference point of a signal transition to *shift* as the programmed swing (the difference between the high and low levels) is varied.

This effect is normally undesirable, and so this timing shift is compensated by an automatic timing correction. The compensation algorithm uses a linear approximation to the slew rate value. This is the *50% Reference* mode of operation.

In some applications, it may be desirable to change a signal's levels *without* affecting its timing. The automatic correction feature can be disabled by selecting the Freeze Reference option in the Level Setup window's Select menu.

When Freeze Reference is selected, the *last* reference correction value used in the 50% Reference mode is stored. All further timing measurements use this same correction value. If 50% Reference is selected once more the timing is corrected to the *actual* level settings. The default setting is 50% Reference.

Note   The comparators need no compensation. The timing trigger reference is independent of the level setting.

Index

HEWLETT PACKARD

Reorder No.
E1280-90222