XILINX®

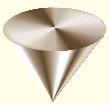
ONLINE

HARDWARE DEBUGGER

REFERENCE/USER GUIDE

TABLE OF CONTENTS

INDEX



GO TO OTHER BOOKS

XSTEP™

Contents

Chapter 1

Introduction

The Hardware Debugger is a graphical user interface that allows you to download a design to a device, verify the downloaded configuration, and display the internal states of the programmed device. Use the program to do the following tasks:

Download a BIT file to an FPGA, or a PROM file to a daisy chain.

- Verify the configuration data of a single device using an XChecker cable.
- Debug the internal logic states of a configured device using an XChecker cable.

The Hardware Debugger replaces the XChecker program.

Refer to the "Programming a Device or a Daisy Chain" chapter for information on how to open, download, and verify your design. Refer to the "Debugging a Device" chapter for information on how to read back and debug your configured device.

Note: Before invoking the Hardware Debugger, you must connect the configuration cable to your PC and target board; you must also be aware of the design and hardware issues described in the section, "Design and Hardware Considerations" in this chapter.

Features of the Hardware Debugger

After creating and implementing a design, you can use the Hardware Debugger to download, verify, and debug.

Downloading refers to the process of programming or configuring a device.

Verification consists of reading the configuration data that was sent to the device and comparing it to the original design to ensure that the design was received by the device.

Debugging consists of reading internal device states to verify that the design is functioning correctly. When using the XChecker cable, the Hardware Debugger can interrogate and display the internal nodes of an FPGA device that are listed in Figure 1-1.

Table 1-1 Probe Points in FPGA Devices

XC4000	XC2000/XC3000/XC5000
CLB ouputs	CLB ouputs
IOB outputs	IOB outputs
RAM/ROM outputs	—

The following is a summary of the Hardware Debugger features:

- Configuration of one or more devices.
- Verification of configuration data for single devices.
- Debugging a single device in synchronous mode or asynchronous mode.
- Generation of multiple graphical and textual waveforms for specific signals and specific device states.
- Specification of signal groups for debugging.
- Saving of waveforms for later use.
- Reuse of debugging settings from one session to another.
- Support for commands from the Console window command bar.
- Creation of macros, that is command scripts, by copying commands from the Console window into a macro window and saving them into macro files.

Design and Hardware Considerations

Before using the Hardware Debugger, you must be aware of the requirements summarized in the following table as early as the

design entry phase. These requirements are dealt with in detail in the chapters “Design Preparation” and ”Connecting Your Cable.”

Table 1-2 Requirements for Hardware Debugger Operations

	Downloading	Verification	Debugging
File Types	BIT, RBT, PROM	BIT, LL	
Design symbols for XC4000/ XC5200 devices	none	READBACK STARTUP *	
Configuration/ readback options for XC2000/ XC3000 devices	Enable pull-up resistor on DONE/PGM pin	**Enable Readback**	
		Never Once X On Command X Enable pull-up resistor on D/P pin	
Configuration/ readback options for XC4000/ XC5200 devices	Enable pull-up on DONE pin	**Readback Control**	
		Readback Clock: X CCLK (for XChecker) User Clock X Capture CLB and IOB Outputs When TRIG Goes Active X Abort Readback When TRIG Goes Inactive	
Cable Type	Any	XChecker	
Pins Used	VCC GND CCLK D/P (DONE) DIN PROG (XC4000) INIT (XC3000/XC4000) RST	VCC GND CCLK RT RD	VCC GND CCLK RT RD TRIG* CLKI * CLKO*
Configuration Mode	Slave	N/A	N/A

*Optional.

Design Entry and Bitstream Generation

You will need access to certain signals depending on whether you generate a design for downloading only or for debugging too.

- For downloading only, use any of the three configuration data file types (BIT, RBT, or PROM) and any download cable. You do not need any special symbols or translation options to generate the configuration data.

Note: To create a PROM file using multiple device configuration data files, use the PROM File Formatter. For more information on how to use the PROM File Formatter, see the *PROM File Formatter Reference/ User Guide*.

- For single device verification or debugging, use a BIT file generated with readcapture enabled from a design that includes the READBACK symbol for XC4000/XC5200 devices only, and optionally, the STARTUP symbol. In addition, you need an XChecker cable.

Hardware Considerations

Before using the Hardware Debugger, you need to make some decisions about the hardware you will be using.

Configuration Mode

When using the download cables to configure a device or daisy chain, you must set the configuration mode of the devices being configured to slave serial. Refer to the *Development System User Guide* for information on how to set the mode pins.

Target Board Selection

Your target board can be either a Xilinx FPGA demonstration board or your own PC board. The demonstration boards can be used to test most designs.

- If you will be using an FPGA demonstration board, set up the board as explained in the "Calc Tutorial" chapter in this guide or in the *Hardware & Peripherals User Guide.*

  You must set the board switches and connect the appropriate cable.

- If you will be using a PC board, you must configure the board and connect the appropriate cable. Refer to the "Connecting Your Cable" chapter for information.

Single or Multiple Device Configuration

You may configure one device or a daisy chain:

- To configure a single device, connect the pins as specified for that particular device type. See Table 4-1 in the "Connecting Your Cable" chapter for information on pin connections.
- For a daisy chain of devices, connect the pins of the lead device for downloading as specified in Table 4-1 in the "Connecting Your Cable" chapter. Connect the pins of the slave devices to the lead device as described in the *The Programmable Logic Data Book*.

Cable Connections

You can perform three main operations using the Hardware Debugger. Each operation requires a specific setup and, in the case of verification and debugging, a particular download cable. Refer to Table 4-1 in the "Connecting Your Cable" chapter for cable connections information for each operation.

Downloading

Connect the cable header connector to your cable assembly and to the configuration pins of your target board. If you have an XChecker cable, connect a cable header connector to the outermost slot of the XChecker cable assembly.

Verification

For this operation, you must use an XChecker cable. Connect both header connectors.

Debugging

Determine which debugging mode, synchronous or asynchronous, you will use and attach the XChecker header connectors accordingly.

Getting Started

This chapter explains how to invoke the Hardware Debugger. It describes the main screen and the three major tasks you can complete: download a design, verify configuration data, and debug the states of a configured device. In addition, it outlines the design and hardware requirements for using the Hardware Debugger.

Starting and Exiting the Hardware Debugger

The Hardware Debugger is a Windows application and can be invoked several ways.

If the Hardware Debugger has been installed as a stand-alone application, the Hardware Debugger icon is present in the XACT*step* program group in the Program Manager. To invoke the Hardware Debugger, double-click on the Hardware Debugger icon in the XACT*step* program group. After invoking the program, use the Open command from the File menu to open the desired configuration file.

You can also invoke the Hardware Debugger from within the Design Manager. To invoke the Hardware Debugger on a specific design, select the desired revision in the Project View, then click on the Hardware Debugger button in the Tools panel. The configuration file is loaded into the Hardware Debugger automatically.

Figure 2-1 Hardware Debugger Icon

Note: Before invoking the Hardware Debugger, you must connect the configuration cable to your PC and target board; you must also be aware of the design and hardware issues described in the section, "Design and Hardware Considerations" in the "Introduction" chapter.

To exit the Hardware Debugger, select the Exit command from the File menu. If you have an open waveform window, you are asked whether you want to save the data before quitting the application.

Using the Hardware Debugger Interface

The Hardware Debugger interface consists of the elements described in this section.

Main Window

The main window, shown in Figure 2-2, is the background against which all windows are displayed. By default, the main window displays a title bar, a menu bar, a toolbar, and a status bar. A control panel appears once you select a debugging mode. You can hide the toolbar, status bar, and Control panel by selecting the Toolbar, Status Bar, or Control panel commands from the View menu.

Title Bar

The title bar displays the program name followed by the path of the currently loaded design.

Menu Bar

The menu bar, located at the top of the Hardware Debugger window, includes the File, Cable, Download, Debug, View, Window, and Help menus. You can also select menu commands by typing the letter underlined in the menu name while holding down the Alt key. Refer to the chapter titled "Command Reference" for information on the menu commands.

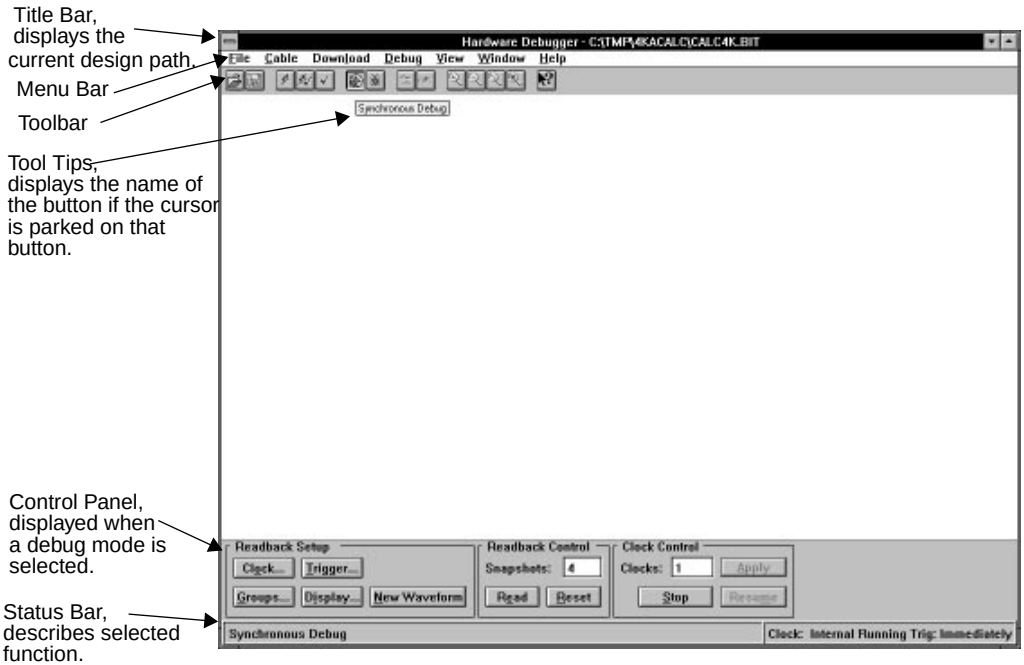


Figure 2-2 Hardware Debugger Main Window

Toolbar

The toolbar is located below the menu bar. It displays several buttons, which you can use to specify commands directly. The toolbar offers an alternative to the menu commands. To get a short description of a toolbar button, park the mouse pointer over the button. After about 2 seconds, a small pop-up text box appears giving more information about that button. This feature is called Tool Tips. The Tool Tips feature displays the name of the button function while the status bar provides similar information.

Control Panel

The Control panel, which is displayed once you select a debugging mode, consists of buttons and fields that you can use to control the active debugging session.

Status Bar

The status bar, located at the bottom of the Hardware Debugger window, provides command and processing information.

Commands and Dialog Boxes

You communicate with the Hardware Debugger by selecting commands from the menu, the toolbar, or the Control panel. Alternatively, you can enter commands from the Console window command bar. Most commands display dialog boxes on which you specify information and options.

All dialog boxes have an OK button and a Cancel button. Moreover, most dialog boxes also have a Help button, a Network button, or both.

- Click on the **OK** button to exit the dialog box and process the information defined on the dialog box.
- Click on the **Cancel** button to exit the dialog box without processing the information defined on the dialog box.
- Click on the **Help** button to get context-sensitive help for the current dialog box.
- Click on the **Network** button to access your network drives. This button is available on Windows for Workgroups installations and is active only for certain PC networks such as Novell.

There are three types of command dialog boxes: file open/file save dialog boxes, filter dialog boxes, and selection dialog boxes.

File Open/File Save Dialog Boxes

The standard file open and file save dialog boxes allow you to load a configuration data file, a saved waveform, or a saved macro; and to save a waveform or a macro. This type of dialog box includes a drive, directory, and file browser as shown in Figure 2-3.

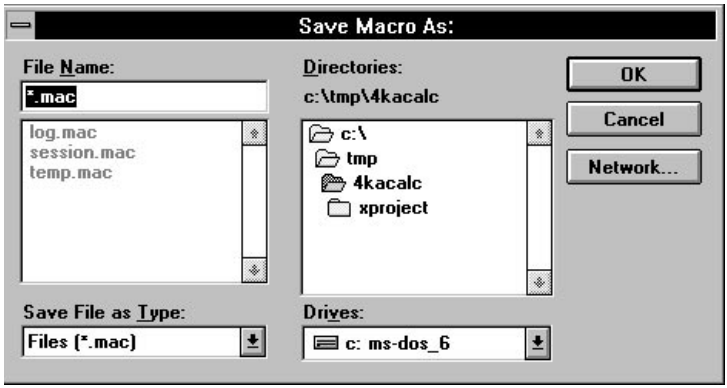


Figure 2-3 Standard File Open/File Save Dialog Box

Filter Dialog Boxes

Filter dialog boxes allow you to specify criteria to select signals and groups for debugging. An example of filter dialog box is shown in Figure 2-4.

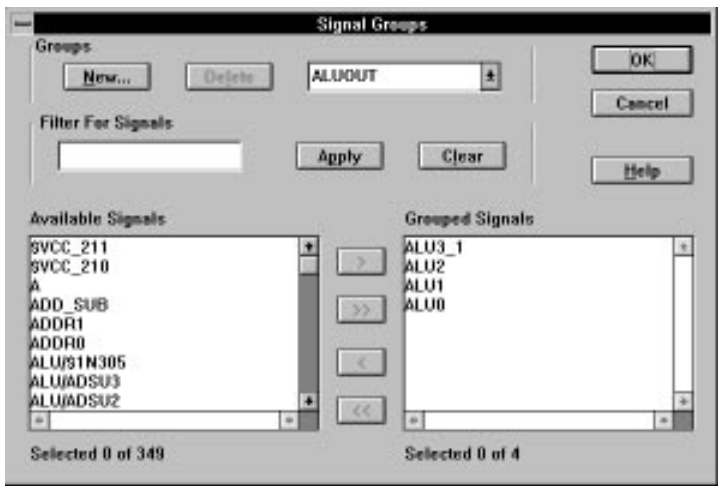


Figure 2-4 Standard Filter Dialog Box

To use the Filter dialog boxes, follow these steps.

1. Specify the pattern of the signal names to include in your display list by typing the characters in the text box located in the Filter for Signals group.

   The characters can be alphanumerical or blank characters.

2. Include one or more wildcard characters (*) to do a global search on the specified string.

   Precede the character string with a wildcard to retrieve all signal names that end the string of specified characters.

   Append the wildcard to the character string to retrieve all signal names that start with the specified character string.

3. Click on the **Apply** button after specifying the Filter criteria.

   The available signals list displays only the signals that match the selection criteria.

4. To clear the filter, click on the **Clear** button or backspace over the information specified in the filter text box.

Selection Dialog Boxes

Selection dialog boxes allow you to specify specific values and selections. An example of selection dialog box is shown in Figure 2-5.

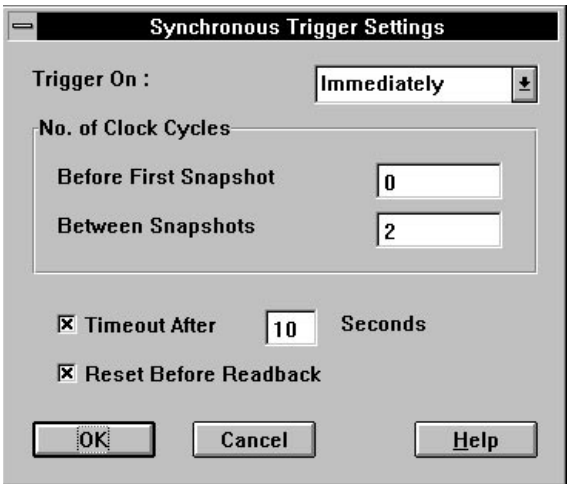


Figure 2-5 Standard Selection Dialog Box

Selecting Commands and Dialog Box Options

To choose a menu item, a toolbar button, or a dialog box option, you can use the mouse or the keyboard.

Using the Mouse

1. Move the mouse cursor over the object you want to select: a toolbar button, a menu option, or a dialog box option; then, click on the left mouse button to select the object.

   If you clicked on a toolbar button, a list box or a dialog box appears. If you clicked on a menu, menu options are displayed and you must select a menu option.

2. To exit a dialog box without selecting anything, select the Cancel button or double-click on the close box in the upper left corner of the dialog box.

3. To get help, click on the Help button on the dialog box.

Using the Keyboard

You can use the keyboard to select objects on your screen, such as a dialog box button or a menu option.

1. To select a dialog box option, use the **Tab** key to position the cursor on that object and highlight it. Press the **Enter** key to process the selection.

   To exit a dialog box without selecting anything, press the **Escape**. key.

2. To choose a menu and display its options, press the **Alt** key and the appropriate underlined letter key corresponding to the menu you want. For example press **Alt-F** to select the File menu.

3. Use the arrow keys to scroll down the list of options in a menu or list box. Press **Enter** when the option you want to use is highlighted or, in the case of a menu item, press the underlined letter corresponding to the menu option you want. For example, press the **N** key to select the New command of the File menu.

4. To get help, press the **Alt** key down and type the letter **H**.

Using the Online Help and Tutorial

The Hardware Debugger includes an online help tool that you can invoke from the menu bar, the toolbar, and the dialog boxes. It also includes an online tutorial that demonstrates the program features.

Online Help

You can access the online help Table of Contents by selecting Contents from the Help menu.

Context-sensitive help refers to the online help available from a dialog box that you are currently using, the help available for a menu command, or the help available for a screen object.

Help on Dialog Boxes

To get context-sensitive help for the current dialog box, either click on the dialog box Help button or press down the Alt key while typing the letter H on your keyboard.

Help for Menu Items

To get context-sensitive help for a menu item, press the F1 function key on your keyboard while selecting the menu item.

Object-Click Help

To get help on a screen object, namely a window, a part of the screen, a toolbar or control panel button, follow these instructions.

1. Click on the Help toolbar button, shown in Figure 2-6.

Figure 2-6 Help Toolbar Button

   The help cursor is displayed and the help button is in the button-down mode.

2. Move the help cursor to the object for which you need help.
3. After positioning the help cursor on the appropriate object, press the left mouse button.

   The help tool displays the corresponding Help page and the Help button returns to the button-up state.

Online Tutorial

To access the online tutorial, select Tutorial from the Help menu. The tutorial page is displayed. Follow the instructions on your screen to view definitions, procedures, and demonstrations of the procedures.

Chapter 3

Design Preparation

This chapter discusses how to prepare a design for use with the Hardware Debugger and how to generate the proper configuration files. The first section, "Creating a Design for Use with the Hardware Debugger," covers the special components needed to perform some of the Hardware Debugger operations. The second section, "Generating Configuration Data Files," discusses the various file types used by the Hardware Debugger and the options that must be specified to create them correctly.

Creating a Design for Use with the Hardware Debugger

To create a design that you can use with the Hardware Debugger, follow the design generation instructions for the type of operation you want to complete for your specific device.

Preparing a Downloadable Design

When using the Hardware Debugger to download a design only, you do not need any special components in the design.

Preparing a Design for Verification and Debugging

Follow the steps outlined in the section for the device family you are targeting XC2000/XC3000 or XC4000/XC5200.

XC2000/XC3000 Designs

You achieve both configuration verification and design debugging by probing the device's internal states. This process is known as readback. For XC2000 and XC3000 devices, a readback is initiated

when a Low to High transition is applied to the M0/RTRIG pin. Once the readback has begun, the serial readback data is presented on the M1(RDATA) pin. Because the readback pins are dedicated pins, you do not need any special components in the design.

XC4000/XC5200 Designs

To provide more flexibility, the XC4000/XC5200 readback signals (RTRIG and RDATA) can be assigned to any of the user programmable device pins as well as the M0 and M1 pins. Because the readback signals are user programmable, you must use the READBACK component in the design when using the Hardware Debugger to verify or debug an XC4000 or an XC5200 device.

To prepare the design for verifying or debugging, use the following steps:

1. Include the READBACK macro in the design schematic.
2. Connect IPAD, OPAD, IBUF, and OBUF primitives to the TRIG and DATA pins of the READBACK macro as shown in Figure 3-1. You can then lock the IPAD and OPAD components to any of the user-programmable I/O locations.

Note: If you want TRIG and DATA to correspond to the mode pins M0 and M1, replace the IPAD and OPAD primitives with the special primitives MD0 and MD1.

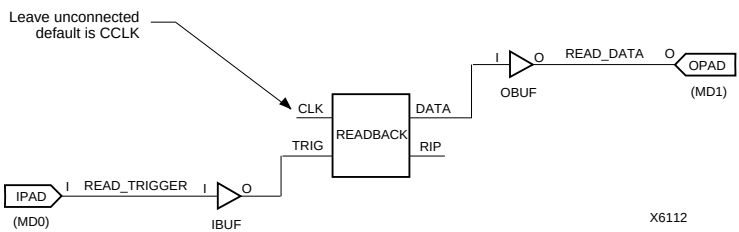


Figure 3-1 READBACK Symbol

3. If you plan on having the Hardware Debugger reset the flip-flops in the design, include the STARTUP symbol and connect the GSR (XC4000) or GR (XC5200) input pin to an unused input of the target device. You will later connect the input to the XChecker cable.

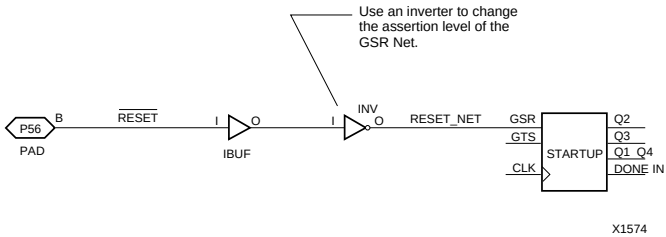


Figure 3-2 XC4000 STARTUP Symbol

Note: Because the XC4000/XC5200 Reset is active High and the Hardware Debugger assumes active Low, the signal sent from the Hardware Debugger through XChecker must be inverted as shown in Figure 3-2 and Figure 3-3.

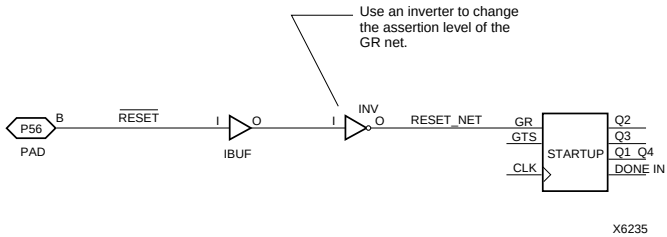


Figure 3-3 XC5200 STARTUP Symbol

The inverter preceding the STARTUP symbol implements the Active-Low Reset asserted by the Hardware Debugger.

Generating Configuration Data Files

Once you have translated the design, you must generate the configuration data files. The Hardware Debugger can download to a chain of multiple devices, known as daisy chain, as well as individual devices. In addition, for single devices, the Hardware Debugger can verify the downloaded configuration and probe the internal states of the device. This process of probing a configured device is also known as readback.

To generate a bitstream, open the implemented design from the Design Manager. If you do not have a project for your design, use the

Design Manager to create a project for that design. If you have a project for your design but the design is not implemented, run the Translate command from the Design menu to read in the changes made in the schematic and to create a new version reflecting the updated schematic, then implement the design as explained in the *Design Manager/Flow Engine Reference/User Guide*.

Creating Files for a Single XC2000 or XC3000 Device

To configure an XC2000 or XC3000 device, you need a bitstream, which can be either a BIT file, an RBT file, or a PROM file. To verify and/or debug a design, you must use a BIT file and have a logic allocation file in your design directory. Use the Design Manager to generate the necessary configuration files.

Creating Downloadable Files

Follow these steps to prepare your XC2000 and XC3000 designs for downloading:

1. From the Design Manager, click the left mouse button on the routed design icon, as shown in Figure 3-4. Then select the **`Implement`** command from the Design menu.

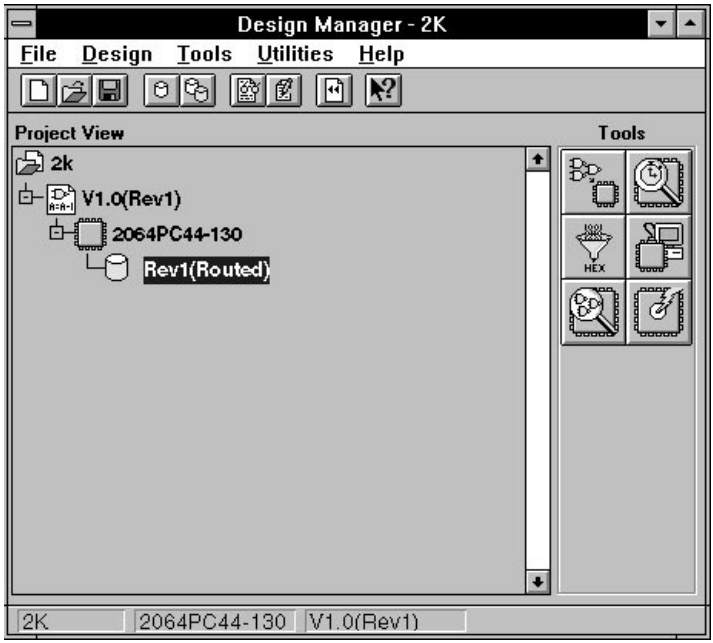


Figure 3-4 Routed XC2000 Design in the Design Manager

The Design Manager Implementation Options dialog box is displayed, as shown in Figure 3-5.

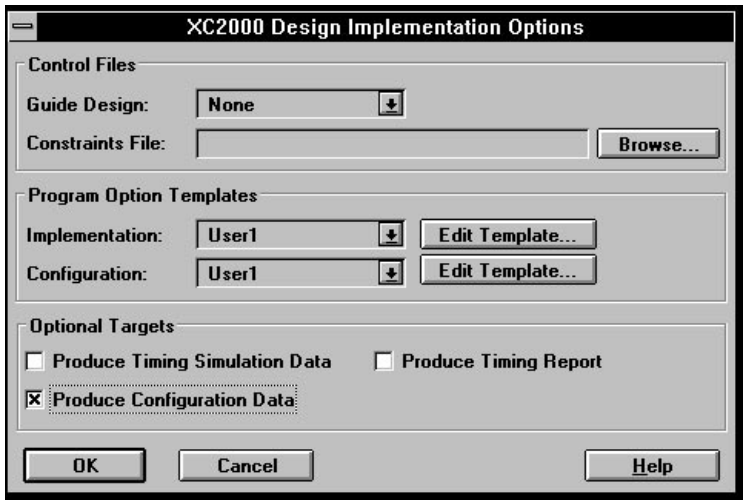


Figure 3-5 Design Manager Implementation Dialog Box (XC2000)

2. Set Configuration Data as the target by clicking in the check box next to **`Produce Configuration Data`** in the Optional Targets group box.
3. Click on the Edit Template button corresponding to the **`Configuration Program Options`** template. The bitstream Configuration dialog box is displayed, as shown in Figure 3-6.

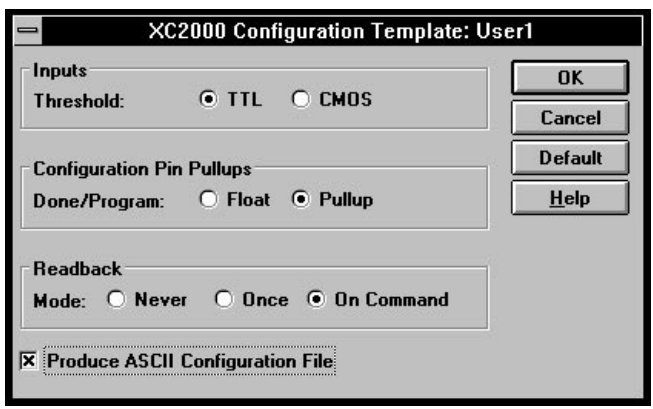


Figure 3-6 Design Manager Configuration Dialog Box (XC2000)

4. Click on **Pull-Up** next to the **Done/Program** pin in the Configuration Pin Pull-Ups group box to enable a pull-up resistor on the D/P pin.
5. Click on **OK** to return to the Implementation dialog box or if you want to enable the readback options, continue with the next section "Creating Files for Verification and Debugging."
6. Select **Run** from the Implementation dialog box to compile the design and produce the configuration data.

Creating Files for Verification and Debugging

Follow these steps to implement XC2000 and XC3000 designs for verification and debugging:

1. Follow steps 1 through 4 in the preceding section "Creating Downloadable Files" to enable a pull-up resistor for the D/P pin for device configuration.
2. Enable the readback capability by specifying the **Readback** option as **On Command** on the Configuration dialog box, as shown in Figure 3-6.

   The software generates a logic allocation file (*design*.ll). The *design*.ll file provides bit locations for the values of I/O, latches, and flip-flops.

Note: For the XC3000 device, select the **Startup/Readback** tab of the dialog box to display the readback options, then set the readback mode as **On Command**, as shown in Figure 3-7.

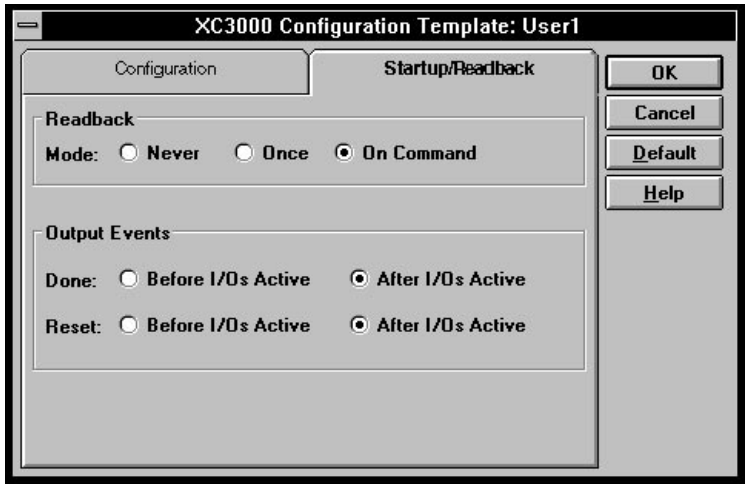


Figure 3-7 Design Manager Configuration Template (XC3000)

3. Click on **OK** to return to the Implementation dialog box.
4. Select **Run** to compile the design and produce the configuration data.

Creating Files for a Single XC4000 or XC5200 Device

To configure an XC4000 or XC5200 device, you need a bitstream, which can be either a BIT file, an RBT file, or a PROM file. To verify and/or debug a design, you must use a BIT file and have a logic allocation file in your design directory. Use the Design Manager to generate the necessary configuration files.

Creating Downloadable Files

Follow these steps to prepare your XC4000 and XC5200 designs for downloading:

1. From the Design Manager, click the left mouse button on the routed design icon, as shown in Figure 3-8. Then select the **Implement** command from the Design menu.

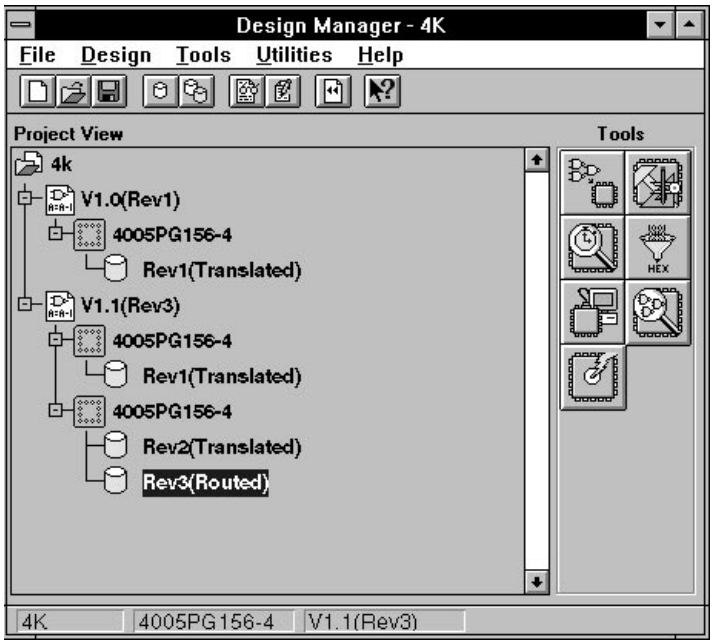


Figure 3-8 Routed XC4000 Design in the Design Manager

The XC4000 Design Implementation Options Dialog Box is displayed, as shown in Figure 3-9.

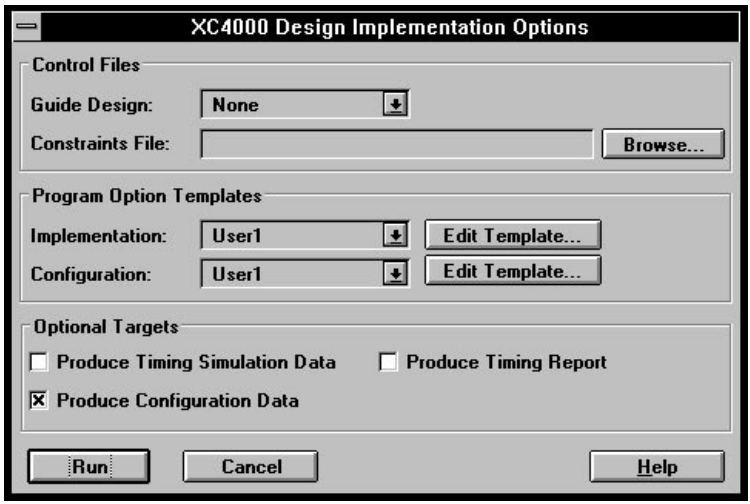


Figure 3-9 Design Manager Implementation Dialog Box (XC4000)

2. Set Configuration Data as the target by clicking in the check box next to **Produce Configuration Data** in the Optional Targets group box.
3. Click on the Edit Template button corresponding to the **Configuration Program Options** template.

   The Configuration Template dialog box is displayed as shown in Figure 3-10.
4. Click on **Pull-Up** next to the DONE pin in the Configuration Pins box to enable a pull-up resistor for the DONE pin.
5. Select **Perform CRC During Configuration** to perform a CRC check of your bitstream during configuration.
6. Select **Produce ASCII Configuration File** to create a raw bits text (RBT) file, which is an ASCII representation of your configuration bitstream.

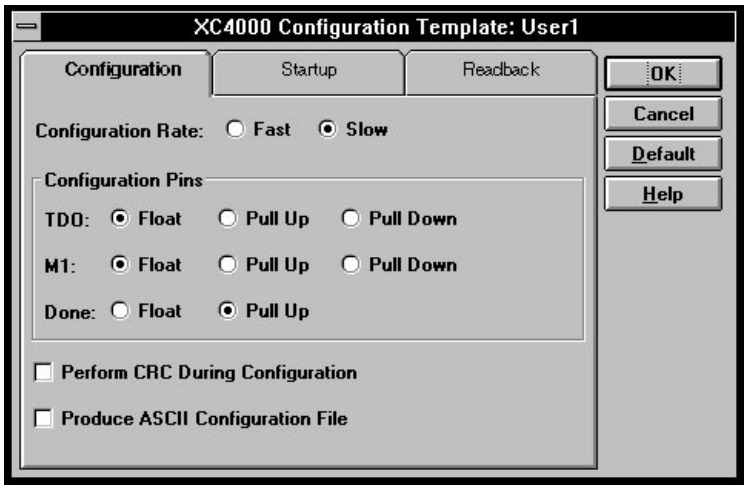


Figure 3-10 Design Manager Configuration Template

7. Click on **OK** to return to the Implementation dialog box or if you want to enable the readback options, continue with the next section "Creating Files for Verification and Debugging."
8. Select **Run** to compile the design and produce the configuration data.

Creating Files for Verification and Debugging

Follow these steps to implement XC4000 and XC5200 designs for verification and debugging.

1. Follow steps 1 through 4 in the preceding section "Creating Downloadable Files" to enable a pull-up resistor for the D/P pin for device configuration.
2. Select the **Readback** tab to access the Readback panel.

   The Readback panel of the Configuration Template is displayed as shown in Figure 3-11.
3. Select **Capture CLB and IOB Outputs TRIG Goes Active** on the Readback panel. This option generates the logic allocation file (*design*.ll). For more information about configuration options,

read the Configuration chapter in the *Design Manager/Flow Engine Reference/User Guide*.

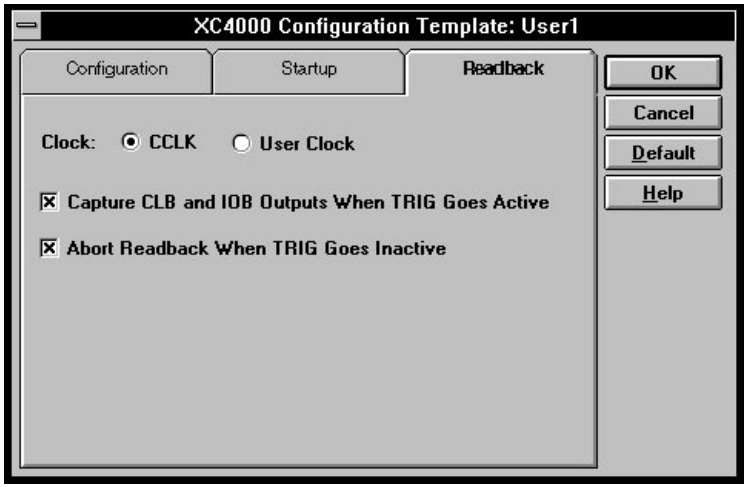


Figure 3-11 Design Manager Readback Data Screen

4. Select **`Abort Readback When TRIG Goes Inactive`** on the Readback panel. This feature allows you to abort a readback in progress.
5. Click on **`OK`** to return to the Implementation dialog box.
6. Select **`Run`** to compile the design and produce the configuration data.

Creating Files for Multiple Devices (Daisy Chains)

To configure a daisy chain of devices, you need a PROM file.

1. Produce the configuration data (BIT files) for each device, referring to the appropriate single device section
2. Concatenate the device BIT files using the PROM File Formatter. For more information, refer to the *PROM File Formatter Reference/ User Guide*. Ensure that the PROM file contains the bitstreams in the same order as the devices on the target board.

Chapter 4

Connecting Your Cable

The Hardware Debugger communicates with your device via a cable. The cable can be a serial, parallel, or XChecker cable. You can use any one of the three cables to download configuration data to a device or daisy chain of devices, but you can only use the XChecker cable to verify and debug.

This chapter describes the various download cables and the operations you can perform with each cable. It also explains how to connect your cable to your PC and target board, and how to set the target board's configuration mode. Additionally, it explains which connections are needed to perform the various Hardware Debugger tasks: download, verify, and debug.

Perform the following steps when connecting your cable:

- Determine the most suitable cable to use based on the tasks you wish to perform.
- Connect the cable to your host system.
- Configure the target board to accept the needed cable connections.
- Connect the cable to your target system.
- Power up the target board.
- Invoke the Hardware Debugger.
- Set the cable options from the Cable menu.

Cable Descriptions

There are three different types of cables: serial, parallel, and XChecker.

Serial Cable

You can use a serial cable only to download configuration data. The serial cable, consists of a cable with a 5-lead header connector with the following connections: V_{CC}, GND, CCLK, D/P, and DIN. It also includes a female RS-232 serial connector.

Parallel Cable

You can use a parallel cable only to download configuration data. The Xilinx parallel cable consists of a 5-lead flying header connector with the following connections: V_{CC}, GND, CCLK, D/P, and DIN. It also includes a 25-pin male connector.

XChecker Cable

You can use the XChecker cable to download, verify, and debug. The XChecker cable assembly houses internal circuitry consisting of a Xilinx FPGA, which functions as an interface between the XChecker software and the target FPGA; a static RAM, which stores the configuration data for download and readback; and an oscillator circuit, which provides a system clock to facilitate download and readback of configuration data.

Using XChecker requires a standard DB-9 or DB-25 RS-232 serial port and may require a DB9/DB25 adapter.

The XChecker cable has 14 signal connections, plus V_{CC} and GND. It comes with two header connectors and two flying lead connectors. See Figure 4-1.

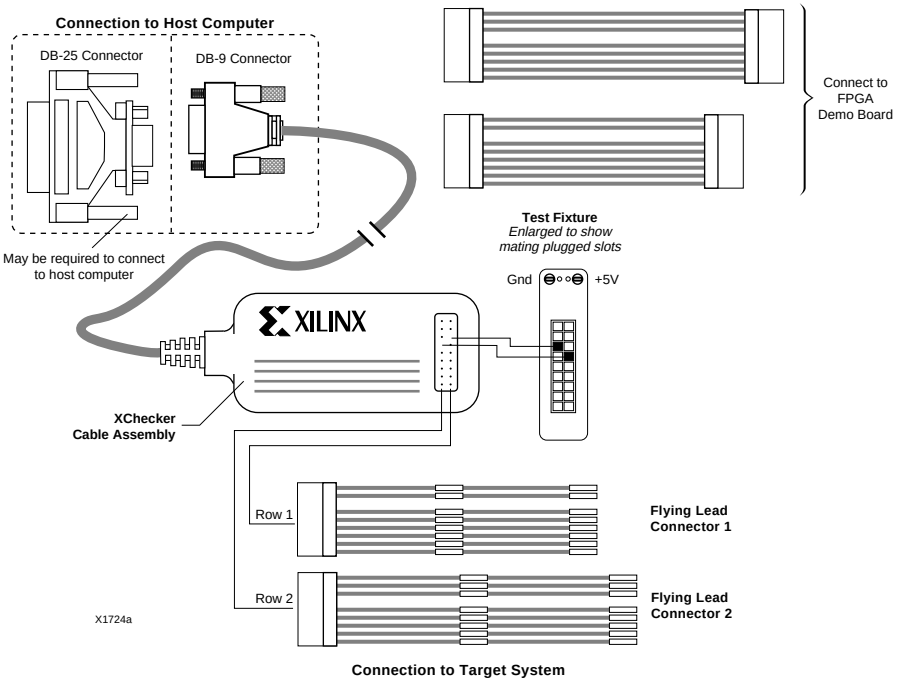


Figure 4-1 XChecker Cable with Connectors and Test Fixture

Connecting the Cable to Your Host System

To install the cable, you must first connect it to the host system.

Parallel Cables

If you have a parallel cable, connect it to the parallel port.

Serial and XChecker Cables

Connect your serial or XChecker cable to your system's RS-232 serial port. A DB-9/DB-25 adapter may be required to connect the cable to your serial port. If you have a different serial port connection, you need to provide an appropriate adapter.

Note: If you are using an XChecker cable with a 3 V adapter, refer to the section "XChecker 3 V Adapter," in Appendix A, for information on how to connect and verify the operation of the 3 V adapter.

Setting Up the Hardware

When using the serial, parallel, or XChecker cables, you must set up the configuration mode of the devices being configured as slave serial. Refer to the *Development System User Guide* or to the *The Programmable Logic Data Book* for information on how to set the mode pins.

Connecting the Cable to Your Target System

This section covers cable connection to the target device. You need appropriate pins on the target system for connecting the target system board to the header connectors on the cable.

Warning: The cable draws its power from the target system through V_{CC} and GND. Therefore, power to the cable, as well as to the target FPGA, must be stable. Do not connect any signals before connecting V_{CC} and GND. The input/output pins of the internal XChecker FPGA should always be at a potential that is lower or equal to their respective rail voltage in order to avoid internal damage.

Serial and Parallel Cables

Connect all the pins of your serial or parallel cable for downloading using the guidelines in Table 4-1 and Table 4-2.

XChecker Cable Connectors

The XChecker cable supports two types of connectors. You can connect to the pins of your target FPGA with a flying lead connector and to the FPGA demonstration boards with a header connector.

- *Flying lead header connectors* have eight standard individual female connectors on one end that fit onto 0.025" square male pins. Each lead is labeled to identify the pin.
- *Header connectors* are standard 9-pin (8 signals, 1 key) header connectors that fit 0.025" square male pins. The pin layout is

shown in Figure 4-2. The header connectors are keyed to ensure that they are properly inserted into the cable assembly.

For each type of connector, there are two different connectors. One is keyed for the downloading signals; the second is keyed for readback signals. Header 1 is the download connector and fits on the outermost connector socket. Header 2 is the readback connector and fits on the inner connector socket. See Figure 4-2.

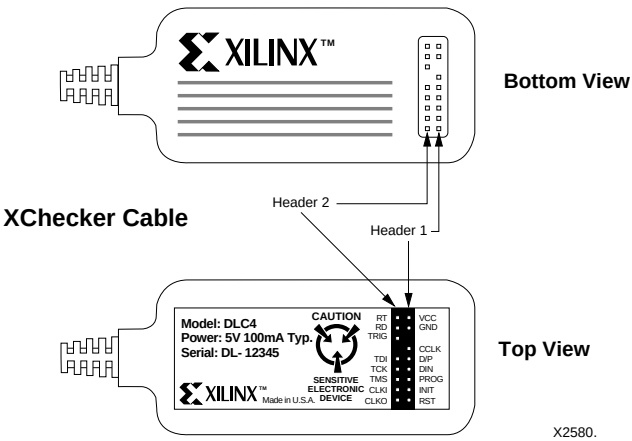


Figure 4-2 XChecker Cable Pins

XChecker Cable Pins

Refer to the tables in this section for information on how to connect the signal pins for specific applications. Table 4-1 shows the necessary connections for each application type, and Table 4-2 describes the pins and how to connect them.

Note: Not all of the signal pins are required for each function.

Table 4-1 XChecker Operation Mode Connections

Cable Header	Pin Name	Download	Verification	Synchronous Logic Probe	Asynchronous Logic Probe
1	V_{CC}	X	X	X	X
	GND	X	X	X	X
	CCLK	X	X	X	X
	D/P	X			
	DIN	X			
	PROG (XC4000 only)	X			
	INIT (XC3000/XC4000 only)	X			
	RST	Opt	Opt	Opt	Opt
2	RT		X	X	X
	RD		X	X	X
	TRIG			Opt	Opt
	TDI				
	TCK				
	TMS				
	CLKI			Opt	
	CLKO			X	

X = Connect as specified in Table 4-2
Opt = Optional.

Refer to Table 4-2 for pin descriptions and connections.

Note: You can connect the system trigger directly to the target FPGA RTRIG pin to latch the state of the device instead of waiting for the XChecker software to initiate the readback.

Table 4-2 Cable Connections and Definitions

Signal Name	Function	XC2000	XC3000	XC4000
V_{CC}	*Power* — Supplies V_{cc} to the cable (5 V, 100 mA, typically)	Connect to the target system V_{CC}		
GND	*Ground* — Supplies ground reference to the cable.	Connect to the target system Ground.		
CCLK*	*Configuration Clock* — Provides a configuration clock to the target system during configuration and readback.	Connect to the target system Configuration Clock. Ensure that all devices are in slave serial mode if you use a download cable to download.		
D/P*	*Done/Program* — Signals the end of configuration. (For XC2000 and XC3000 devices, a High to Low transition on D/P coupled with a High to Low on Reset, causes the device to reprogram.)	Connect to the D/$\overline{P}$ pin with a 10-50 kΩ pull-up resistor.		Connect to the target system DONE pin with a 10-50 kΩ pull-up resistor.
DIN	*Data In* — Provides configuration data to the target system during configuration and is 3-stated at all other times.	Connect to the target system's lead device DIN pin.		
PROG* (XC4000 Only)	*Program* — A 300ns Low pulse causes the device to reprogram. (A Low from the device indicates that the device is clearing its configuration memory.)	N/A		Connect to target system $\overline{PROG}$ with a 10-50 kΩ pull-up resistor.
INIT*	*Initialize* — Indicates the start of configuration for XC3000/XC4000 parts. For XC3000A and XC4000 devices, a logic zero on this pin during configuration indicates a data error.	N/A	Connect to the target system INIT with a 10-50 kΩ pull-up resistor.	
RST*	*Reset* —During configuration, a Low pulse causes XC2000 and XC3000/XC3000A devices to restart the configuration process. After configuration, this pin can drive Low to reset the target FPGA internal latches and flip-flops.	Connect to the target FPGA $\overline{RESET}$ pin with a 10-50 kΩ pull-up resistor.		User-programmable connection; requires a 10-50 kΩ pull-up resistor.

Signal Name	Function	XC2000	XC3000	XC4000
RT*	*Read Trigger* — XChecker output. The Hardware Debugger provides a Low-to-High transition on RT to initiate a readback.	Connect to M0/RTRIG with a 10-50 kΩ pull-up resistor.		User-programmable connection; requires a 10-50 kΩ pull-up resistor.
RD*	*Read Data* — XChecker input. The Hardware Debugger receives the readback data through the RD pin once a readback has been initiated.	Connect to M1/RDATA through a 10-50 kΩ pull-up resistor in slave serial configuration mode.		User-programmable connection; requires a 10-50 kΩ pull-up resistor.
TRIG*	*System Trigger* — XChecker input. A High on this pin signals the XChecker electronics to initiate a readback and causes the RT pin to go High.	Connect to the target system readback trigger and to an external pin if you are using an external signal to trigger the readback.		
TDI TCK TMS	Reserved.	N/A		
CLKI	*Clock Input* — Transmits your system clock to the XChecker electronics. This clock must be between 120 kHz and 10 MHz. Connect this pin to the target system clock to synchronize the readback trigger with the target system clock.	Connect to the source of the target system clock for synchronous debugging.		
CLKO	*Clock Output* — Drives the target system clock. The clock can come from either the CLKI pin, or it can be internally generated by the XChecker cable when CLKI is unconnected.	Connect to the destination of the target system clock for synchronous debugging.		

*You can sample the signals marked with an asterisk and display their logic states by using the **`Logic Level of Header Pins`** option of the Cable menu.

Note: XChecker does not drive the configuration mode pins (M0, M1, M2) during configuration. You must specify the logic levels for these pins externally.

Connecting for Download

To connect your cable for downloading only, connect your configuration cable to your target FPGA device. Refer to Table 4-1 and Table 4-2 for pin assignment information.

Connecting for Verification

You can use XChecker to verify a previously configured FPGA. Refer to Table 4-1 for pin assignment information.

Connecting RT and RD

Make sure to connect the XChecker RT and RD pins to the FPGA RTRIG and RDATA pins, respectively. If you used the symbols MD0 and MD1, consult the device pinout tables in *The Programmable Logic Data Book* for the exact locations of M0 and M1. If you used IPAD/OPAD primitives, consult the I/O Pin Assignments Report available from the Design Manager's Report Browser.

Connecting for Synchronous Debugging

In synchronous mode debugging, you can control the target FPGA clock through the XChecker cable. You can control the number of clock pulses applied and the frequency at which the clock cycles occur. For synchronous debugging, connect the TRIG, RT, RD, CLKO and, optionally, CLKI pins of the XChecker cable to the target FPGA, as shown in Table 4-1 and Table 4-2. This XChecker cable configuration allows you to download, verify, and debug your design in the synchronous mode.

Connecting the XChecker Clock

To allow the XChecker clock to control the target FPGA system clock, connect XChecker's CLKO pin to the pin that you assigned as the FPGA's clock pin. All the synchronous logic should be connected to this clock source to do synchronous debugging.

The source of the CLKO clock signal can come either from an internally generated XChecker clock pulse or from an external clock pulse that you provide.

- To use the XChecker internal clock, connect the CLKO pin to the target FPGA. Leave the CLKI pin unconnected. Then, select the internal clock setting in the Hardware Debugger software.
- To use an external clock, connect the user clock to the XChecker CLKI pin and connect the XChecker CLKO pin to the FPGA. Then, select the external clock setting in the Hardware Debugger software.

Connecting an External Trigger

To use an external trigger, such as the terminal count of a counter or some other condition in your target board to initiate a readback, make sure to connect the external trigger signal to the XChecker TRIG pin.

You do not need the TRIG signal if you plan to use an internal trigger, such as the Enter key on the keyboard to initiate a readback.

Connecting RT and RD

Make sure to connect the XChecker RT and RD pins to the FPGA RTRIG and RDATA pins, respectively. If you used the symbols MD0 and MD1, consult the device pinout tables in *The Programmable Logic Data Book* for the exact locations of M0 and M1. If you used IPAD/OPAD primitives, consult the I/O Pin Assignments Report available from the Design Manager's Report Browser.

Connecting for Asynchronous Debugging

This configuration allows the target system to run while a readback is executed. There is no need to provide the system clock to XChecker, as readback is executed independently of the system clock. Connect the system clock so that it controls the device flip-flops directly. Thus, the CLKI and CLKO pins on the XChecker cable are not used for asynchronous debugging.

Connect the external trigger and the RD and RT pins as explained in the previous section "Connecting for Synchronous Debugging."

Setting the Cable Options

After connecting the cable for the functions you need (download, verification, or debug), power your target board to enable the software to communicate with the XChecker cable, and invoke the Hardware Debugger. You must then set the cable options.

1. Select the **`Communications`** option from the Cable menu to invoke the Communication Setup dialog box, shown in Figure 4-3.
2. Specify the cable type, baud rate, and port settings for downloading and debugging.
3. Click on the **`OK`** button to accept the selections and close the dialog box.

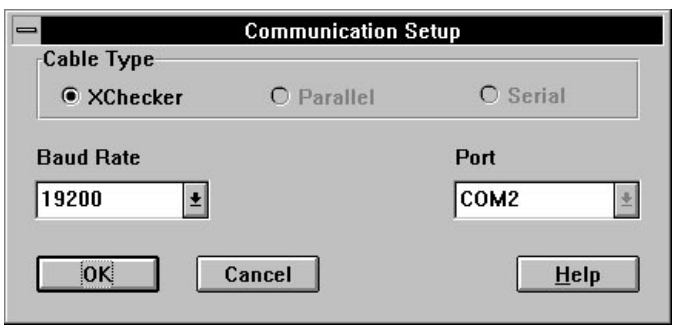


Figure 4-3 Communication Setup Dialog Box

Cable Type

Select the cable type you have installed for downloading.

Baud Rate

Specify a communications baud rate between the cable and the host system. You cannot specify the baud rate for a parallel cable.

Communication between the host system and the XChecker cable is dependent on host system capability. Valid baud rates include 9600, 19200, and 38400.

Port

Specify the port to be used for downloading and readback. This list box contains a list of valid ports for the platform. If you selected a serial cable or an XChecker cable, the serial ports are displayed (COM1, COM2...). If you selected a parallel cable, the parallel ports are displayed (LPT1, LPT2...).

Resetting the Cable

Use the Reset command of the Cable menu to reset the internal logic of the cable after a power glitch.

Note: You may need to reconfigure your target device after a complete loss of power to your target board.

1. Select the **Reset** option of the Cable menu to reset the internal logic of the cable.

   The cable is reinitialized and the proper baud rate is set. A dialog box displays information about the cable type, port connection, and baud rate.

2. Check the cable settings displayed in the dialog box.

Chapter 5

Programming a Device or a Daisy Chain

The Hardware Debugger enables you to program the logic of a device. In this chapter, you will be using the Download menu commands to configure one or more devices. You are first shown how to verify the cable connections and how to open a design file. Then, you learn how to download a design file to your target device, how to verify the logic of the design, and how to configure multiple devices.

Downloading is the process of programming a target device with the logic functions contained in the design you download. Verification is the process of reading back the downloaded configuration data and comparing it to the original data to ensure that the data was downloaded correctly.

Before completing any of the above tasks, turn on the power to your board.

Warning: You must connect V_{CC} and Ground before you connect the control signals to XChecker. The input/output pins of their internal FPGA should always be at a potential that is lower or equal to their respective rail voltage in order to avoid internal damage.

Preparing for Download and Verification

When you invoke the Hardware Debugger, the cable you have installed is automatically detected. To modify the Communications settings, refer to the section “Setting the Cable Options” in the previous chapter “Connecting Your Cable.”

Checking the Cable

It is good practice to check the logic levels of your header pins before programming or reading back a device to ensure that you have connected the cable properly.

Note: Do not connect the XChecker leads to signals of different voltage levels than required as you might damage the XChecker internal hardware.

- Select **`Logic Level of Pins`** from the Cable menu to display the logic levels of the cable pins and to check that the pins are connected properly.

Opening a Design File

After powering your target board, invoking the Hardware Debugger, and setting the cable options, you are ready to open and download a design file. You can open the following types of files: BIT files, RBT files, or PROM files.

- BIT files and RBT files contain data for a single device. Use the Design Manager to create them.
- PROM files (.mcs, .tek, or .exo) contain data to program daisy chains. Use the PROM File Formatter to create them.

Note: You can only use a BIT file when verifying and debugging.

1. Select **`Open Bitstream`** from the File menu or select the Open Design File button from the toolbar.

Figure 5-1 Open Design File Toolbar Button

The Open Design File dialog box is displayed, as shown in Figure 5-2.

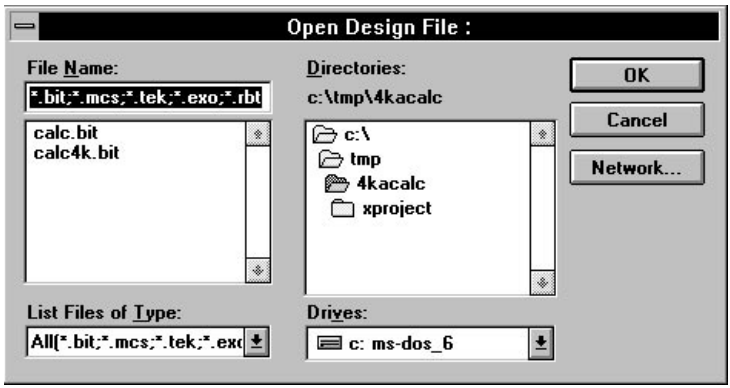


Figure 5-2 Open Design File Dialog Box

2. In the Drives box, select the desired drive. If the drive you want is not listed, type the desired drive name in the File Name box alone or as part of the file name.
3. In the Directories box, select the desired directory.
4. In the List Files of Type box, specify the extension for the type of file you want to open. This extension is then displayed in the File Name box. Only the files that match this extension are displayed in the File Name box.
5. In the File Name box, type the name of the design file you want to open. The extension is optional. By default, the .bit extension is used.
6. Click on **OK**.

Note: If you do not have a logic allocation file (*design*.ll), a pop-up dialog box appears to inform you that debugging is not possible. The design.ll file is created during the implementation process if you have selected the appropriate readback option. Click on **OK** to proceed.

Once you have opened your design, the path and file name of your design are displayed in the title bar at the top of the Hardware Debugger screen.

Downloading a Design to a Target Board

Once you have set up the appropriate cable and you have opened a file to download, you may download the configuration data to a target board.

Note: If you plan on verifying or debugging a design, download a BIT file. You cannot verify or debug PROM files or RBT files. For more information, see the chapter "Design Preparation."

1. Select **Reset** from the Cable menu to reset the cable.
2. Select **Logic Level of Header Pins** from the Cable menu to verify the logic levels of the XChecker cable, in particular the state of the D/P pin should be Low.

   If necessary, make changes to the pin configuration and repeat steps 1 and 2.
3. Select **Download Design** from the Download menu or select the Download button from the toolbar.

Figure 5-3 Download Toolbar Button

A download status window, shown in Figure 5-4, is displayed. This window includes a status bar that is updated as the data blocks are transmitted.

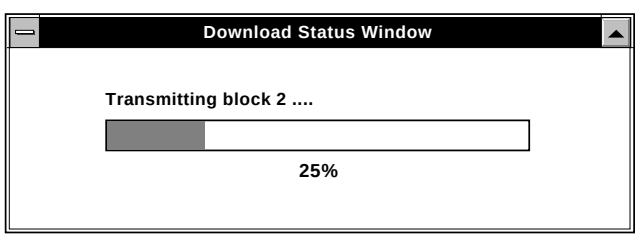


Figure 5-4 Download Status Window

If the design was downloaded successfully, a message appears informing you that the DONE signal went high and provides you

with the transmission time. Finally, the number of mapped bits is displayed in a dialog box.

Note: Alternatively, you can use the Download and Verify option provided you have connected the XChecker readback and verification pins to your target board. The design is verified only if the current file is a BIT file configured for readback, and if you are using an XChecker cable connected for verification. Refer to Table 4-1 and Table 4-2 in the "Connecting Your Cable" chapter for information. To download and verify, you can select the Download and Verify button from the toolbar.

Figure 5-5 Download and Verify Toolbar Button

Verifying Design Logic

The Verify Bitstream command verifies a design that you have downloaded using a BIT file.

Note: To verify and debug a device configured from a BIT file, you must have a logic allocation (design.ll) file in the design directory, enable the Readback option (for XC4000 designs, enable the Readback Capture option), use the XChecker cable, and, if you have an XC4000 design, you must use the READBACK symbol.

1. Select **`Verify Bitstream`** from the Download menu or select the Verify button from the toolbar.

Figure 5-6 Verify Toolbar Button

The device's configuration is read back and compared to the downloaded data to ensure that the device was properly configured.

When verification is completed, a message appears to let you know the number of verified bits.

2. Check that the number of verified bits corresponds to the number of downloaded bits.

Configuring Multiple Devices

You can use a PROM file to program several FPGAs at once.

Configuration

To configure a daisy chain of devices, connect the XChecker cable to the lead device and ensure that all the devices in the daisy chain are connected correctly.

1. Connect the device pins of the master device as specified for the operation you want.
2. Connect the DOUT output of each slave device to the DIN input of the next device in the chain. Refer to *The Programmable Logic Data Book* for more information on how to connect devices in a daisy chain.
3. Open a PROM file by selecting **`Open Bitstream`** from the File menu.
4. Select **`Download Design`** from the Download menu or select the Download button shown in Figure 5-3 from the toolbar.

   The Download Status window, shown in Figure 5-4, is displayed. This window includes a status bar that is updated as the data blocks are transmitted.

   If the design was downloaded successfully, a message appears informing you that the DONE signal went high and provides you with the transmission time. Finally, the number of mapped bits is displayed in a dialog box.

Verification

The Hardware Debugger and XChecker cable cannot read back or verify a daisy chain of FPGA devices. They can only read back and verify individual devices in a daisy chain.

Chapter 6

Debugging a Device

Debugging refers to the process of reading back the internal states of a configured device using an XChecker cable to ensure that the device is behaving as expected. After simulating a design to test the design using worst-case delays, debug the device to analyze its real-time behavior.

This chapter discusses how to debug a configured FPGA in either synchronous or asynchronous mode and how to save the readback data in a file.

Debugging a Configured FPGA

After configuring a device, you can analyze its behavior by taking snapshots of the device's probe points. Flip-flops, RAMs, CLB outputs, and IOBs are all probe points and are read back when a snapshot of the device is taken.

There are two debugging modes for capturing the states of a device: synchronous and asynchronous.

During synchronous debugging, you check the device behavior in conjunction with a clock controlled by the XChecker cable. A controlled clock enables you to control which states are captured. To control the clock, you can use the XChecker internal clock, as opposed to a system clock or a device clock. The XChecker cable can also interface with a system clock, enabling you to start, stop, and restart the clock.

During asynchronous debugging, the Hardware Debugger does not control the clock and therefore does not control which states are captured. You can only select the signals that you want to display in your waveform.

Debugging Overview

This section summarizes the debugging process. For more details, continue with the section "Synchronous Mode Debugging" or the section "Asynchronous Mode Debugging."

To debug XC4000 and XC5000 designs, you must have the READBACK symbol in your design. Also add the STARTUP symbol if you plan to reset your design using the Reset button located on the Control panel in the Hardware Debugger.

All devices require that you have a logic allocation file, *design*.ll, in your design directory. This file defines the probe points or storage elements, such as, the RAM locations, flip-flops, and I/Os.

After you download a BIT file using the Download menu command, you can read back the states of the configured device using the Debug menu options.

1. Select a debugging mode from the Debug menu, Synchronous Mode or Asynchronous Mode.
2. Set the appropriate options, such as the trigger type, the signals to display, and, in the case of synchronous debugging, the clock options, using the Settings submenu under the Debug menu.
3. Open a new waveform window by clicking on the New Waveform button in the Control panel.
4. Read the signals that you selected for display.

   The signals are displayed in the active waveform window.

Note: You can view up to a maximum of 450 snapshots in a single waveform. To view snapshots beyond that number, you must open a new waveform window.

Debugging a Previously Debugged Design

To debug a previously debugged design, specify the debug mode after opening the design. The Hardware Debugger loads the relevant settings for the specified debugging mode. The data, summarized in Table 6-1, is saved in a design-specific .ini file in your working directory. If the saved settings data is incomplete, the Readback button is disabled. Defaults are supplied for Trigger and Clock

Settings. You must still specify the signals to be displayed to enable the Readback button.

Table 6-1 Available Sync and Async Settings

Settings	Synchronous Mode	Asynchronous Mode
Trigger	X	X
Number of clock cycles before and between snapshots	X	
Timeout	X	X
Reset before readback	X	X
Clock	X	
Number of clocks	X	
Snapshots	X	
Groups and their radix settings	X	X
Displayed signals	X	X

Synchronous Mode Debugging

You must first download your design before you can debug it. The synchronous mode requires an interface, the XChecker cable, which enables you to do a controlled readback of your design. In this debug mode, you are controlling the clock through the XChecker cable.

The snapshot and clock features, which are available when performing synchronous mode debugging, enable you to define specific clock patterns to gather the device states you want. For example, if you are debugging a counter, you may want to check that the combinational values generate the state value of 10. Set the Number of Clocks Before First Snapshot to 10 and the number of snapshots to 1 to capture state 10 of the device.

Note: To verify, read back, or debug a BIT file, you must have a logic allocation (*design*.ll) file in the design directory, enable the Readback option (for XC4000 designs, enable the Readback Capture option),

use the XChecker cable, and, if you have an XC4000 or XC5000 design, you must use the READBACK symbol.

In addition, you can use these options:

- Resume Clock: This option establishes a free-running clock and advances the states of the device forward.
- Stop Clock: This option stops the clock being applied on the CLKO pin.
- Reset FPGA: This option sets the device to the zero state.
- Apply Clocks: This option moves the device forward the number of states set by the Number of Clocks to Apply option.

Pin Assignments

Before enabling the Synchronous Mode, decide whether you will use an internal or an external clock.

- To use the internal XChecker clock, connect the CLKO pin to the FPGA system clock input pin. Leave the CLKI pin unconnected.
- To use an external clock, connect the system clock to the XChecker CLKI pin and connect the XChecker CLKO pin to the FPGA system clock input pin.

Debugging in the Synchronous Mode

To debug in the synchronous mode, follow these steps.

1. Select **Synchronous Mode** from the Debug menu or select the Synchronous Debug button from the toolbar.

Figure 6-1 Synchronous Mode Toolbar Button

The Control panel, shown in Figure 6-2, is displayed and the Settings commands of the Debug menu are enabled.

Figure 6-2 Control Panel (Synchronous Debugging)

2. Select **`Settings` → `Clock`** from the Debug menu or click on the Clock button on the Control panel.

   The Clock command invokes the Clock Settings dialog box.

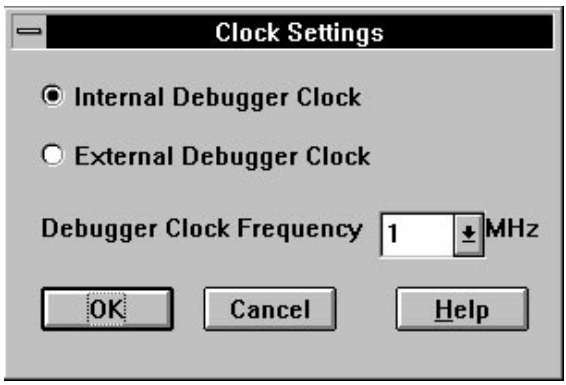


Figure 6-3 Clock Settings Dialog Box

3. Click on the appropriate radio button for the clock you will be using: **`Internal Clock`** or **`External Clock`**.

   - If you select an Internal Clock, set the Debugger Clock Frequency (1, 3, 5, or 11 MHz).

     For 3 Volt devices, the only allowable speed is 155 KHz; therefore, it is the only speed shown in the Frequency box.

   - If you select the external clock, the FPGA uses the system clock connected to the XChecker CLKI pin. In this case, you cannot set the clock speed. The list box for the Clock Frequency is disabled.

4. After setting the desired clock settings, click on **OK**.

   The Clock Settings dialog box is closed and the clock settings are displayed on the status bar.

5. Select **Settings → Trigger** from the Debug menu or click on the Trigger button on the Control panel.

   The Trigger command invokes the Synchronous Trigger Settings dialog box, which enables you to select the type of trigger you wish to initiate the readback.

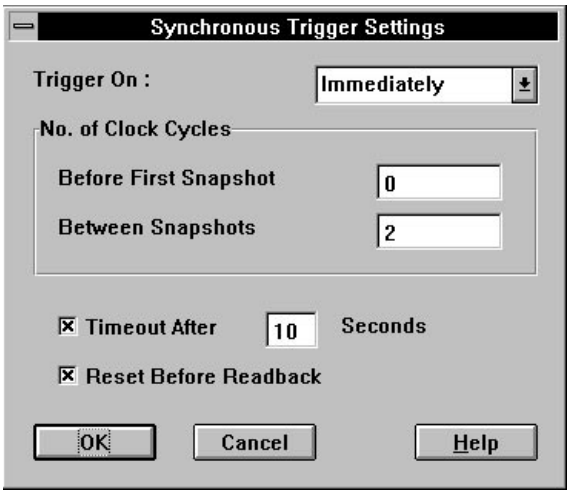


Figure 6-4 Synchronous Trigger Dialog Box

- Trigger On

  You may choose to use the **External Pin** (TRIG on the XChecker cable) or the **Enter Key** to initiate a readback, or you may wish to initiate a readback **Immediately** after the Read FPGA command is invoked.

- Number of Clock Cycles

  Specify the number of clock cycles to be applied before the first snapshot and between snapshots. You use these options to cycle the device before the first snapshot and between multiple snapshots.

- Timeout After

  Use this option to specify the cutoff time for a trigger to be detected. If the trigger is not received within the specified time, the Read FPGA command is aborted.

- Reset Before Readback

  Use this option to reset the device each time you execute the Read FPGA command.

6. After setting the desired trigger settings, click on **OK**.

   The Synchronous Trigger Settings dialog box is closed and the Trigger settings are displayed on the status bar.

7. Select **Settings → Display Signals** from the Debug menu or click on the Display button on the Control panel.

   The Display Signals command invokes the Display Signals dialog box, shown in Figure 6-5, which enables you to choose the signals and signal groups you wish to view.

8. Use the Filter for Signals box to select several signals whose names match the pattern you enter in the box. Refer to the section "Generating a Signal List" for details.

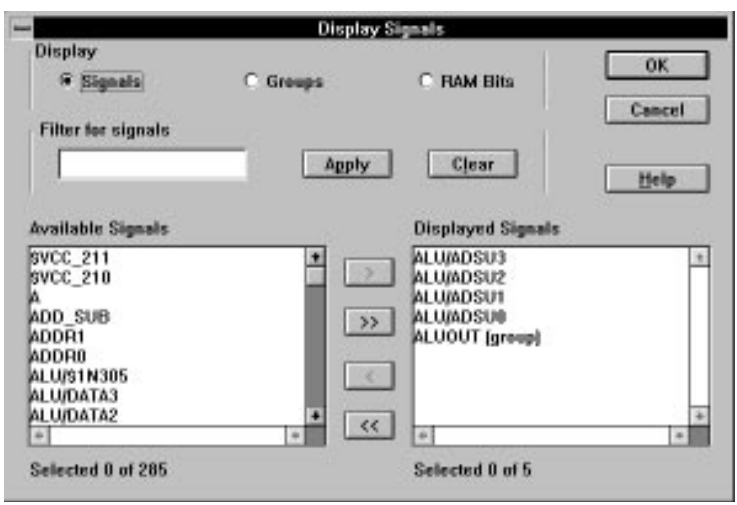


Figure 6-5 Display Signals Dialog Box

9. After selecting the desired signals and groups to be displayed, click on **OK**.

   The Display Signals dialog box is closed and the selected signals are displayed on the status bar.

10. Select **Settings → Number of Snapshots to Read** from the Debug menu. Alternatively, you can type a number in the Snapshots text box on the Control panel.

    The Snapshots Count dialog box is displayed, as shown in Figure 6-6, allowing you to enter the number of snapshots to read.

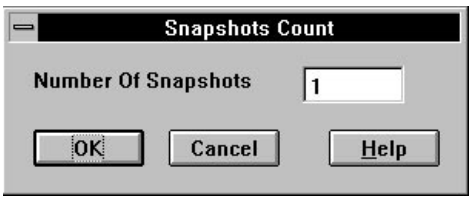


Figure 6-6 Snapshots Count Dialog Box

11. After setting the desired number of snapshots, click on **OK**.

    The Snapshots Count dialog box is closed and the Number of Snapshots is displayed on the status bar.

12. Select **Read FPGA** from the Debug menu or click on the Read button in the Control panel to read the states of the signals that you selected for debugging.

    The device being read back blindly returns its configuration data and the state of every probe point when a readback is triggered.

    The software then extracts the signals you selected and displays the signal values in a waveform.

Note: The readback data stream is linked to the Active Waveform window only. Once the connection is closed, the waveform window will no longer be updated with new information. The maximum number of snapshots you can view in a single waveform is 450.

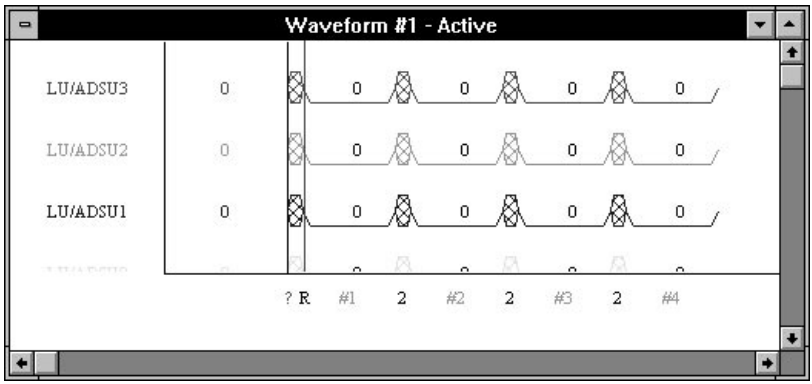


Figure 6-7 Active Waveform Window

Cycling the Device

Part of controlling the clock through the XChecker cable consists in applying the number of clocks you want to your device. When you apply clocks to your device, you advance the state of your device.

Use the Apply Clocks command to apply the number of clocks specified in the Number of Clocks to Apply box.

1. Select **`Settings → Number of Clocks to Apply`** from the Debug menu and enter the number of clocks to apply or type a number in the Clocks text box in the Control panel.

   When you select the Apply Clocks command, the Hardware Debugger uses the clock number you specified in this dialog box to cycle the device.

2. Select **`Apply Clocks`** from the Debug menu to cycle the device the number of clocks specified in the Number of Clocks to Apply box. Alternatively, you can click on the Apply button on the Control panel.

When the number of clocks to apply is set to 1, the Apply Clocks command can be used to single step the device during synchronous debugging.

Resetting the FPGA

If you have not set the Reset Before Readback option, choose **Reset FPGA** before issuing the Read FPGA command.

Viewing Additional Signals

To view new signals on your waveform, you must add these signals to the display list and read the device states again.

1. Use the **Display Signals** command or select Display on the Control panel to add new signals to the display list.
2. Add the new signals to the display and select **Read FPGA** to read the states of the new signals.

Viewing the Waveform in Text Mode

Select **Text Data** from the View menu to display the snapshots in text form.

Asynchronous Mode Debugging

Asynchronous mode debugging allows you to use any external clock. Because the XChecker cable does not control the system clock, any number of clocks can occur between snapshots; thus, all snapshots are asynchronous to the system clock.

Note: To verify, read back, or debug a BIT file, you must have a logic allocation (*design*.ll) file in the design directory, enable the Readback option (for XC4000 designs, enable the Readback Capture option), use the XChecker cable, and, if you have an XC4000 design, you must use the READBACK symbol.

Pin Assignments

In asynchronous mode, you do not use the XChecker cable to control the clock; therefore, you must leave the XChecker CLKI and CLKO pins unconnected. Instead, you use the system clock. Connect the system clock so that it controls the target device flip-flops directly.

Debugging in the Asynchronous Mode

In asynchronous mode debugging, you use a free-running clock to cycle the device.

1. Select **`Asynchronous Mode`** from the Debug menu or select the Asynchronous Debug button from the toolbar.

Figure 6-8 Asynchronous Mode Toolbar Button

The Control panel, shown in Figure 6-9, is displayed and the Settings menu options are enabled.

Figure 6-9 Control Panel (Asynchronous Mode)

Note: Because the Hardware Debugger does not control the system clock, the clock and snapshot settings are disabled.

2. Select **`Settings`** → **`Trigger`** from the Debug menu or click on Trigger on the control panel.

   The Trigger command invokes the Asynchronous Trigger Settings dialog box, which enables you to select the type of trigger you wish to initiate the readback.

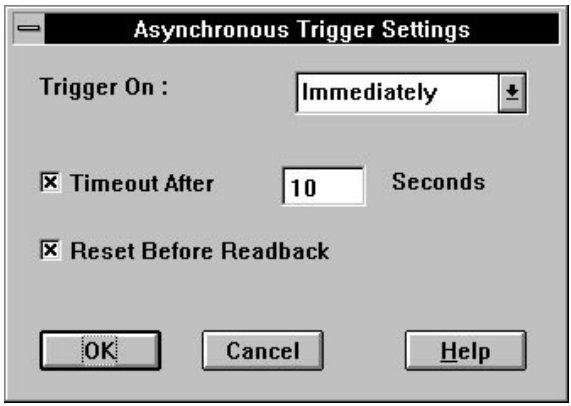


Figure 6-10 Asynchronous Trigger Settings Dialog Box

- Trigger On

  You may choose to use the **External Pin** (TRIG on the XChecker cable) or the **Enter Key** to initiate a readback, or you may wish to initiate a readback **Immediately** after the Read FPGA command is invoked.

- Timeout After

  Use this option to specify the cutoff time for a trigger to be detected. If the trigger is not received within the specified time, the Read FPGA command is aborted.

- Reset Before Readback

  Use this option to reset the device each time you execute the Read FPGA command.

3. After setting the desired trigger settings, click on **OK**.

   The Asynchronous Trigger Settings dialog box is closed and the Trigger settings are displayed on the status bar.

4. Select **Settings → Display Signals** from the Debug menu to choose the signals and groups that you wish to display or click on the Display button on the Control panel. For more details on how to select signals for display, see the section "Generating a Signal List" in this chapter.

5. Choose **Reset FPGA** from the Debug menu or click on the Reset button on the Control panel whenever you need to reinitialize the device.
6. Select **Read FPGA** from the Debug menu to read the states of the signals that you selected for debugging or click on the Read button on the Control panel.

   The device being read back blindly returns its configuration data and the state of every probe point when a readback is triggered.

Note: The readback data stream is linked to the Active Waveform window only. Once the connection is closed, the waveform window will no longer be updated with new information. The maximum number of snapshots you can view in a single waveform is 450.

Creating a Signal Group

Use the Settings → Signal Groups command from the Debug menu to create a group of signals, or to add or remove signals from a previously defined group of signals. Use this option in conjunction with the Display Signals command when you are in the process of debugging a design.

1. Select **Settings → Signal Groups** from the Debug menu.

   The Signal Groups command invokes the Signal Groups dialog box, which enables you to select the signal groups you wish to probe.

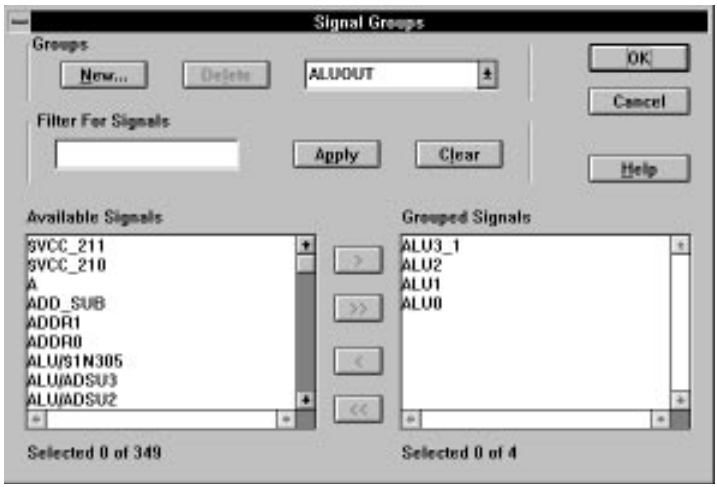


Figure 6-11 Signal Groups Dialog Box

2. Click on the **New** button and type the name of the group you want to create in the Name text box on the Group Name pop-up dialog box.
3. Select the signals you want to include in the group by highlighting the desired signals in the Available Signals list box and selecting the single right arrow button to move the highlighted signals to the Grouped Signals list box.

   You can remove signals from the group in the same manner by selecting the single left arrow button.

   To add or remove all the signals displayed in a list box, use the double-arrow buttons.
4. You may use the **Filter for Signals** box to make signal selection easier. For example, if you specify the letter A followed by the wildcard character '*', you will filter out all signal names that do not start with the A character.

   Click on **Apply** to apply the filter to the Signals displayed in the Available Signals list box. To redisplay the complete signal list, click on the **Clear** button.

5. Click on the **OK** button to save the new group and exit the Signal Groups window.
6. Refer to the "Generating a Signal List" section for information on how to include the groups you created in your display list.

Modifying a Group

The Signal Groups dialog box contains a list of previously defined signal groups and a list of signals. This dialog box enables you to add or remove signals from a group, provided the group has not been included in the list of displayed signals.

Note: If a group is included in the Displayed Signals list, you must remove it from the display list before you can modify it.

Removing a Group from the Display List

To remove a group from the display list, use the Display Signals dialog box.

1. Select **Settings → Display Signals** from the Debug menu or click on the Display button on the Control panel.
2. Click on the **Groups** radio button on the Display Signals dialog box.

   Groups that are included in the display list are listed in the Displayed Signals column.
3. Click on the group you want to modify in the Displayed Signals list and click on the left arrow button (<) on the dialog box to remove the group from the list of included signals.
4. Click on the **OK** button to close the Display Signals dialog box.
5. Select **Settings → Signal Groups** from the Debug menu or click on the Groups button on the Control panel.
6. Click on the down-arrow of the Groups pull-down list box and select the desired group.

   The signals currently included in the selected group are listed in the Group Signals list box. The signals that are not included are displayed in the left hand side list box.
7. Add or remove the signals you want using the arrow buttons.

8. Click on the **OK** button to save the new group and exit from the Signal Groups window.

Note: To delete a group, go to the Signal Groups dialog box and select the existing group from the Groups pull-down list box. Then click on the Delete button.

Generating a Signal List

The signal list can consist of signals, groups, and RAM bits. To create the list of signals to be displayed, including any signal groups that you have created, use the Display Signals dialog box.

Note: To display a new signal group, you must first define it in the Display Groups dialog box before adding it to the display list. Refer to the procedure "Creating a New Signal Group."

Creating a List of Signals to Display

Invoke the Display Signals dialog box and select which signals to include in the display list. The display list is a list of signals, groups, and RAM bits you select for probing with the Hardware Debugger.

1. Select **Display Signals** from the Debug Settings submenu to display the Display Signals window or click on the Display button on the Control panel.

   The Display Signals dialog box, shown in Figure 6-12, is displayed.

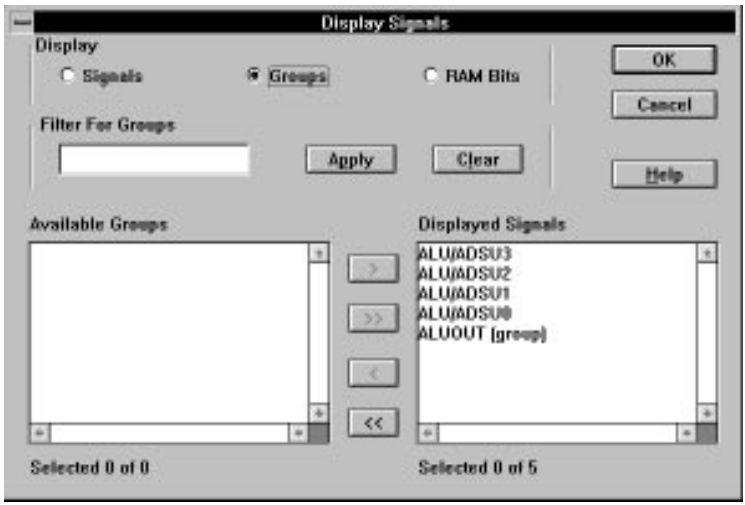


Figure 6-12 Display Signals Dialog Box

2. Click on **`Signals`**, **`Groups`**, or **`RAM Bits`** to display the list of available signals, signal groups, or RAM bits.
3. In the Available Signals list box, highlight one or more of the signals, groups, or RAM bits you want to probe.
4. Click on the single right arrow button to move the highlighted signals to the Displayed Signals list box.

   You can remove signals from the display list in the same manner by selecting the signals and clicking on the single left arrow button.

   To add or remove all the signals displayed in a list box, use the double-arrow buttons.
5. You may use the **Filter for Signals** box to make signal selection easier.

   For example, if you specify the letter A followed by the wildcard character '*', you will filter out all signal names that do not start with the A character.
6. Click on **`Apply`** to apply the filter to the Signals displayed in the Available Signals list box.

To redisplay the complete signal list, click on the **Clear** button.

7. Click on the **OK** button to exit from the Display Signals dialog box.

   The current waveform is automatically updated with the new signals, which appear as cross-hatched hexagons.

Saving and Loading Readback Data

After generating a readback sequence of your signals, you may save the readback data you generated. Saving readback data is useful for multiple design analyses, for comparison, and design optimization.

The Hardware Debugger enables you to view previously saved readback data textually or in waveforms. When the file is first opened, the data is displayed in a waveform. To view the data textually, use the Text Data option of the View menu.

Note: From the Hardware Debugger, you can only re-open waveforms that were saved as graphical windows. To save readback data for later use, view it as a graphically waveform first, then save it as a graphical waveform with the .rdb extension.

Saving Readback Data

1. From the File menu, choose **Save Readback** or select the Save Readback button from the toolbar.

Figure 6-13 Save Readback Toolbar Button

The Save Readback Data dialog box, shown in Figure 6-14, is displayed.

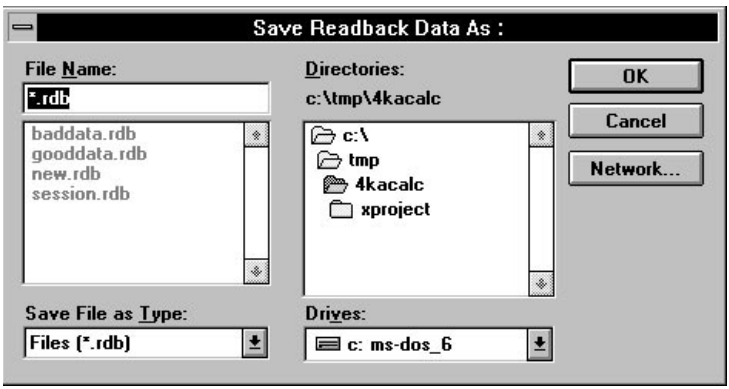


Figure 6-14 Save Readback Data Dialog Box

2. In the File Name box, type or select the name of the file you want to save. By default, the .rdb extension is appended to the file name.

   If you wish to change the path for your readback data, you can do the following:

   - In the Drives box, select the desired drive. If the drive you want is not listed, type the desired drive name in the File Name box or as part of the file name.
   - In the Directories box, select the desired directory.

3. Click **OK** to save the data.

Viewing Previously Saved Readback Data

Use the Open Readback Data command from the File menu to view previously saved readback data that was saved as a graphical waveform.

Note: You cannot open from the Hardware Debugger a waveform window that was saved in the text format.

1. Use the **Open Readback Data** option from the File menu to view previously saved readback data.

2. Specify the file name and press the **OK** button.

   The data is displayed in the format in which it was saved, as a graphical waveform.

3. To view the data textually, select the **Text Data** option under the View menu.

   The waveform is displayed in text format. See Figure 6-15.

4. To switch the data back to the graphical mode, select the **Waveform** option under the View menu.

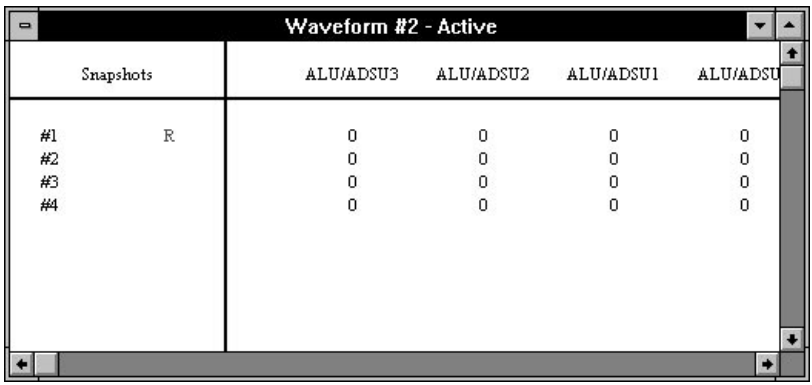


Figure 6-15 Text Data Window

Chapter 7

Customizing the Interface

This chapter discusses how to customize the Hardware Debugger. You can optimize its use by creating macros using pre-defined settings and changing the display parameters. Customization tasks are outlined as follows:

- Using Macros
- Controlling the Waveform Window

Using Macros

The Hardware Debugger allows you to define macros to automate tasks. After you have completed a procedure, you can capture the steps you want to include in your macro by selecting the commands from the Console window, copying them into a macro window, and saving the macro.

Alternatively you can type the commands directly in a macro window. For information about the available commands and their specific syntax, refer to the "Console Commands" appendix.

Creating a Macro

Follow these steps to create a macro:

1. Select **`New Macro`** from the File menu.

   A macro window is opened and the macro toolbar is displayed, as shown in Figure 7-1.

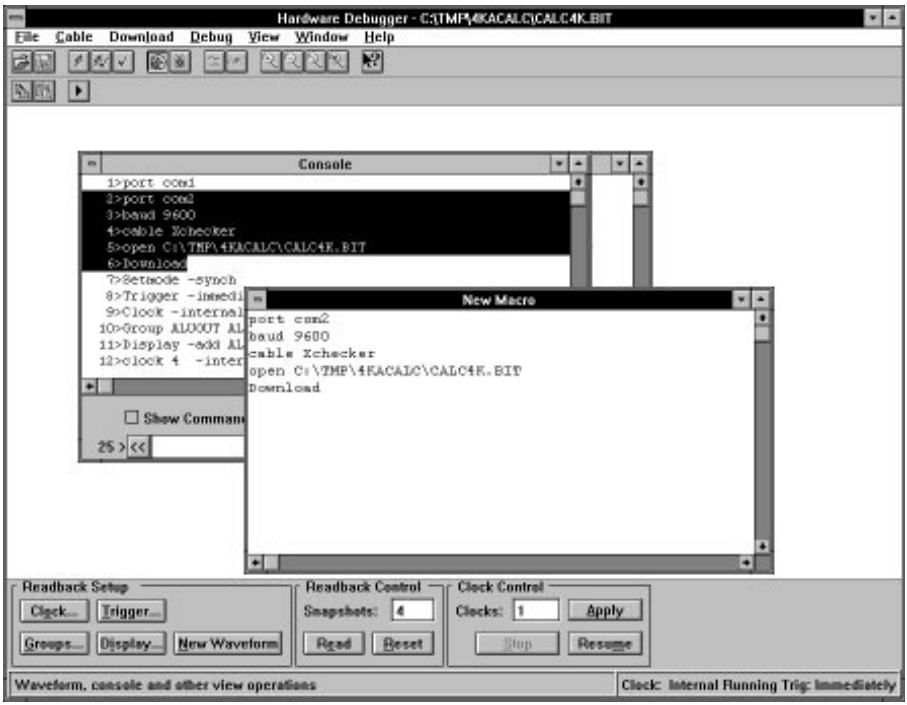


Figure 7-1 New Macro Window

2. Select **`Console`** from the View menu to invoke the Console window and view the commands that have been executed.
3. The Console window is displayed as shown in Figure 7-2.
4. Click the left mouse button at the beginning of the first command you want to copy.
5. Holding down the Shift key, click the left mouse button at the end of the last command you want to copy.

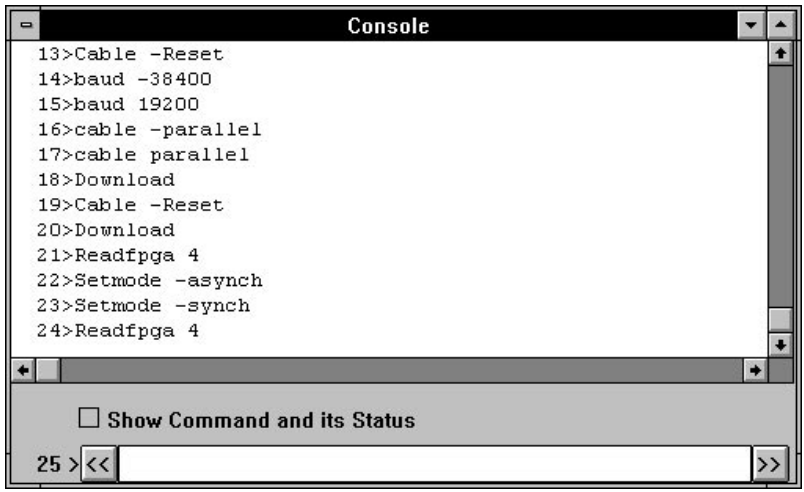


Figure 7-2 Console Window

6. Copy the steps you want from the Console window using the **Copy** button from the macro toolbar.

Figure 7-3 Copy Macro Toolbar Button

7. Select the macro window.
8. Paste the commands you have copied into the macro window using the **Paste** button from the macro toolbar.

Figure 7-4 Paste Macro Toolbar Button

9. Repeat steps 2 and 3 to copy non-consecutive lines into the macro.
10. Save the macro using the **Save Macro** command in the File menu.

Editing an Existing Macro

To edit an existing macro:

1. Select **Open Macro** from the File menu and open the macro you wish to modify from the list of macros.
2. Select commands by double-clicking the left mouse button to select a word. You can also select an entire block of text by pressing the left mouse button, dragging the mouse to the end of the region you wish to copy, and then releasing the mouse button.
3. Use the **Copy** and **Paste** buttons from the macro toolbar to copy the selected information and paste it at the new insertion point.
4. Select the **Save Macro** command from the File menu to save the changes.

Running a Macro

To run a macro, follow these steps:

1. Open the macro by selecting **Open Macro** from the File menu.
2. Select the **Run** button from the macro toolbar.

Figure 7-5 Run Macro Toolbar Button

Saving the Console Log to a File

After invoking commands, you can check the command history for a debug session by opening the Console window. You can also save the command history, or command log, to a text file.

1. Select the contents of the Console window by clicking on the left mouse button at the top of the window and dragging the cursor down to the end of the window.

   To select the entire log, select the top part of the log, then click on the scroll bar to scroll down to the bottom of the page. At the bottom of the page, press the Shift key down while you click on the mouse button again to complete the selection.

2. Press **Ctrl-C** on your keyboard to copy the selected contents of the Console window to the clipboard.
3. Invoke a Windows text editor, such as Microsoft Write, open a new file, and paste the contents of the clipboard into the file by pressing **Ctrl-V** on your keyboard.

   The contents of the Console window appear inside the file.
4. Save the file using the appropriate text editor command.

Controlling the Waveform Display Parameters

This section describes the features available for customizing your waveform window. After generating a waveform, you can modify its format by changing the display mode from graphics to text, modifying the radix of your groups and the color or the position of your signals. To display the waveform textually, select the Text Data option from the View menu. In addition, you can change the size of the waveform and the position of the windows on the screen.

Description of the Text Window

In the text mode, the waveform window displays a textual representation of the device states you specified for viewing.

The text window, displayed in Figure 7-6, is a tabular display of the readback information.

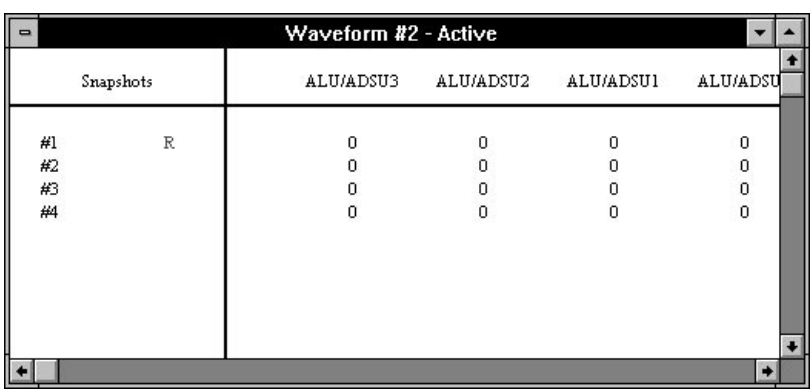


Figure 7-6 The Active Waveform Window (Textual View)

Signals and Signal Groups

The display area shows the readback values of signals and signal groups.

The left-most column lists the snapshots in sequential order. The signals and groups are listed horizontally at the top of the table. The readback value of each signal is listed under the signal and in the row corresponding to the snapshot for which the value was captured.

Reset is indicated by a capital R to the left of the vertical axis.

Vertical Axis

The vertical axis on the waveform separates the snapshots count from the signal names and values.

Signals and Signal Groups

The display area shows the readback values of signals and signal groups.

The left-most column lists the snapshots in sequential order. The signals and groups are listed horizontally at the top of the table. The readback value of each signal is listed under the signal and in the row corresponding to the snapshot for which the value was captured.

Reset is indicated by a capital R to the left of the vertical axis.

Vertical Axis

The vertical axis on the waveform separates the snapshots count from the signal names and values.

Description of the Graphical Window

In the graphical mode, the waveform window displays a graphical representation of the signals you specified for viewing.

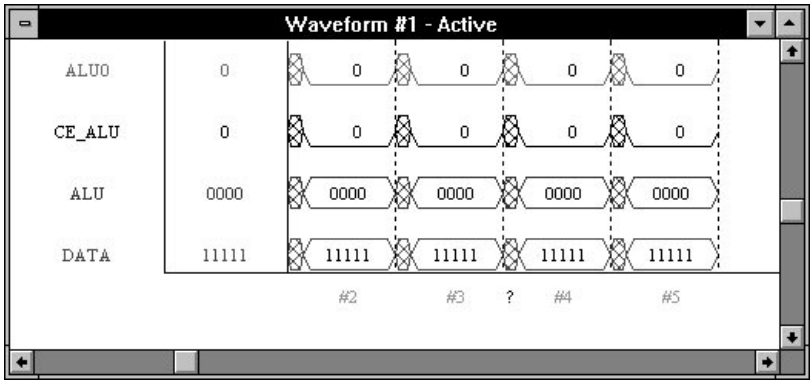


Figure 7-7 The Active Waveform Window (Graphical View)

Signals and Signal Groups

The display area shows waveforms for the readback values of signals and signal groups.

- Each *Signal* is displayed as a semi-hexagonal shape within which the readback value of the signal is displayed.
- Each *Signal Group* is displayed as a complete hexagon within which the concatenated values of all the members of the group are displayed.
- *Readback values* are separated by hexagonal cross-hatched patterns.

Vertical Axes

The two left-most vertical axes on the waveform separate the signal names from their values.

- The signal name column, left of the first vertical axis, shows the last eight characters of the displayed signal names and groups.
- The current value column, left of the second vertical axis, shows the current value of the displayed signals and groups.

A red axis is displayed to indicate a reset. A capital R is also displayed at the point where the reset axis intersects the horizontal axis.

Horizontal Axis

The horizontal axis marks the snapshots (#1, #2, #3...) and the number of clock cycles applied between consecutive snapshots.

- A capital R indicates a reset.
- The first number in a readback series represents the clock number applied before the first snapshot is taken.
- A question mark (?) indicates a free-running clock.

Grid

The horizontal axis uses equispaced marks to indicate snapshots, regardless of the number of clock cycles that might occur in-between them. For each snapshot, a pair of vertical dotted lines representing the span of the readback event is displayed. These pairs of lines constitute the *grid*, displayed in Figure 7-7. To turn on the vertical grid that separates snapshots from one another, select **`Vertical Grid`** from the View menu.

Changing the Size of the Waveform

Use the options listed under the View menu to change the size of the waveform on your screen or to update your screen.

- Select **`Zoom In`** from the View menu to increase the size of your waveform or click on the Zoom In button from the toolbar.

Figure 7-8 Zoom In Toolbar Button

- Select **`Zoom Out`** from the View menu to decrease the size of your waveform or click on the Zoom Out button from the toolbar.

Figure 7-9 Zoom Out Toolbar Button

- Select **`Zoom Region`** from the View menu or click on the Zoom Region button from the toolbar to delimit a region of the waveform that you wish to display .

Figure 7-10 Zoom Region Toolbar Button

- Select **`Zoom Full`** from the View menu or click on the Zoom Full button from the toolbar to display the entire waveform.

Figure 7-11 Zoom Full Toolbar Button

- Select **`Refresh`** from the View menu to redraw the elements on your screen thereby enabling the software to update the current waveform with the latest debugging modifications you might have made.

Controlling the Position of the Windows

To change the way the windows are displayed on your screen, follow these guidelines:

- Select **`Cascade`** from the Window menu to display each window on top of the preceding window in cascade fashion.
- Select **`Tile`** from the Window menu to arrange the windows next to one another within the main window so that all windows are legible. The window that is in focus will process the commands you invoke.
- Select **`Arrange Icons`** from the Window menu to arrange the iconized windows horizontally.

Changing the Radix of Displayed Signal Groups

Use the Change Radix dialog box to display or change the radix used by the Hardware Debugger to annotate your bitstream signal groups with readback signal values.

The Radix group box shows the signal groups and their radices. Possible radices include binary, octal, decimal, and hexadecimal.

You can specify different radices for different signal groups provided you make the changes one after the other.

1. Select **Change Radix** from the View menu.

   The Radix of Groups dialog box is displayed.

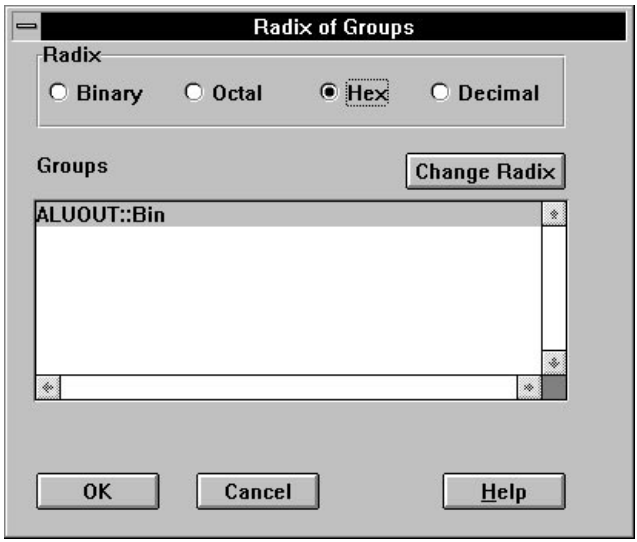


Figure 7-12 Radix of Groups Dialog Box

2. Select the groups whose radices you wish to change.
3. Click on the desired radix.
4. Click on the **Change Radix** button to change the radix for the selected signals in the list box. The following will be displayed:

   signal_name::*radix_name*

5. Click on the **OK** button to exit the Radix of Groups dialog box and update the waveform.

Changing the Color of the Waveform Window

You can change the color of the following waveform elements, listed as menu items: signals, axes, background, and grid.

1. If you want to modify a signal color, click on the specific signal waveform to be modified.

   To modify other waveform elements, skip step one.

2. Select **`Change Color`** from the View menu.
3. From the Change Color submenu, select the item whose color you wish to modify:
   - Signals: This option modifies the color of the selected signal.

     This option is only available if a signal is selected in a Waveform Display window.
   - Axes: This option modifies the color of the current waveform axes.
   - Background: This option modifies the background color of the current waveform.
   - Grid: This option modifies the grid color of the current waveform.
4. Select a new color.
5. Click on **`OK`** to make the color change.

Moving a Signal

Use the Move command from the View menu to move a signal from a location on the Waveform window to another location so as to modify the order in which the signal waveforms are displayed.

1. Select the signal on the waveform.
2. Select **`Move`** from the View menu.

   The cursor arrow icon turns into a large + icon to indicate that a move is pending.
3. Click on the signal waveform after which you wish to insert the selected signal.

You can move signal waveforms one at a time.

Resetting the Color to Its Default

Select **`Reset Color to Default`** from the View menu to use the default colors for the current waveform.

Chapter 8

Menu Command Reference

This chapter lists in alphabetical order the commands that are available from the Hardware Debugger menu bar. When applicable, the appropriate toolbar button is enclosed in the command description.

Menus

The Hardware Debugger has six menus, which are described in the following sections.

File Menu

The File menu contains commands to load a configuration file, save readback data, use macros, print waveforms, and exit the program. To access the File menu with the keyboard, press the letter F, underlined in the menu name, while holding down the Alt key. Press the underlined letter corresponding to the menu option you want. The commands available from the File menu are the following:

Open Bitstream	Opens a configuration data file.
Open Readback Data	Opens readback data saved from a waveform window.
Close	Closes the selected window.
Save Readback	Saves the selected waveform window.
New Macro	Opens a new macro.
Open Macro	Opens an existing macro for you to edit or execute.
Run Macro	Executes the selected macro.

Save Macro	Saves the selected macro window as a macro file.
Save Macro As	Saves the selected macro under a new name.
Print	Prints the selected waveform window.
Print Setup	Sets up the print options.
Exit	Exits the Hardware Debugger.
1, 2, 3, 4	Enables you to select the four last design files that were opened.

Cable Menu

The Cable menu includes commands to set the cable communication between the Hardware Debugger and your target board. To access the Cable menu with the keyboard, press the letter C, underlined in the menu name, while holding down the Alt key. Press the underlined letter corresponding to the menu option you want. The commands available from the Cable menu are the following:

Communications	Specifies the cable options including the cable type, baud rate, and port options.
Self-Check	Performs a diagnostic of the XChecker cable's internal components and data transmission capabilities.
Logic Level of Header Pins	Displays the logic values of the cable pins.
Reset	Resets the logic of the XChecker cable.

Download Menu

The Download menu includes commands to download and verify a device configuration. To access the Download menu with the keyboard, press the letter L, underlined in the menu name, while holding down the Alt key. Press the underlined letter corresponding to the menu option you want. The commands available from the Download menu are the following:

Download a Design	Programs the target device or devices with the configuration data.
Download and Verify	Programs and verifies a target device.
Verify Bitstream	Verifies that a target device has been programmed correctly.

Debug Menu

The Debug menu includes commands to specify the debugging mode and to debug. To access the Debug menu with the keyboard, press the letter D, underlined in the menu name, while holding down the Alt key. Press the underlined letter corresponding to the menu option you want. The commands available from the Debug menu are the following:

Synchronous Mode	Sets synchronous mode debugging.
Asynchronous Mode	Sets asynchronous mode debugging.
Settings	Sets the options for the current debugging mode including the following: clock, trigger, signals, clock number, and snapshot options.
Reset FPGA	Resets the flip-flops of the target FPGA.
Read FPGA	Reads the state of the target device and displays the signals you selected for debugging in a waveform.
Apply Clock[s]	Applies the number of clocks you specified.
Stop Clock	Interrupts the system clock.
Resume Clock	Reapplies the system clock.
New Waveform	Displays a new waveform window.

View Menu

The View menu includes commands to open the Console window, display special screen areas, and customize the waveform. To access the View menu with the keyboard, press the letter V, underlined in the menu name, while holding down the Alt key. Press the

underlined letter corresponding to the menu option you want. The commands available from the View menu are the following:

Console	Toggles the Console window on or off.
Control Panel	Toggles the control panel on or off.
Toolbar	Toggles the toolbar on or off.
Status Bar	Toggles the status bar on or off.
Text Data	Displays the selected readback data in a text waveform.
Waveform	Displays the selected readback data in a graphical waveform.
Vertical Grid	Toggles the vertical grid on or off.
Change Radix	Enables you to modify the radix of your groups.
Move Signal	Changes the position of a signal waveform in the waveform window.
Change Color	Changes the color of the following waveform elements: Signals, Axes, Background, and Grid.
Reset Color to Default	Resets the waveform to the default colors.
Zoom In	Increases the size of the selected waveform.
Zoom Out	Decreases the size of the selected waveform.
Zoom To Region	Focuses on the selected region.
Full	Resizes the waveform to show all data.
Refresh	Redraws the waveform with the latest signals added to the display list.

Window Menu

The Window menu includes commands to change the position of the Hardware Debugger windows on your screen. To access the Window

menu with the keyboard, press the letter W, underlined in the menu name, while holding down the Alt key. Press the underlined letter corresponding to the menu option you want. The commands available from the Window menu are the following:

Tile	Arranges the windows vertically on the screen.
Cascade	Positions the windows on top of one another in cascade fashion so that only the active window is fully shown, and only the title bars of the nonactive windows are visible.
Arrange Icons	Enables you to reposition the windows that were reduced to icons on your screen.
1, 2, 3, ...	Enables you to open the windows that were opened during a program session.

Help Menu

The Help menu contains commands that enable you to access online help for the Hardware Debugger. To access the Help menu with the keyboard, press the letter H, underlined in the menu name, while holding down the Alt key. Press the underlined letter corresponding to the menu option you want. The commands available from the Help menu are the following:

Contents	Lists online help topics.
Search for Help On	Searches for specific topics.
Tutorial	Invokes the interactive tutorial.
About Hardware Debugger	Displays the program version.

Alphabetical Listing of Commands

The following commands are available in the menus of the Hardware Debugger's graphical user interface.

1, 2, 3, 4 (File Menu)

The File menu displays up to four of the most recently opened design files. To open any one of these files, click on the desired file in the menu.

1, 2, 3, ... (Window Menu)

The Window menu lists the names of all windows you have opened during a program session. These windows can be waveform windows, the Console window, or macro windows. To open one of these windows, click on the desired window in the menu.

About Hardware Debugger (Help Menu)

Select this command to display the Hardware Debugger version.

Apply Clock[s] (Debug Menu)

Apply *Control Panel Button*

Select this command to single step the clock during synchronous debugging. Each time you click on the Apply button, the number of clocks you specified in the Number of Clocks to Apply field is applied. This option is available only after you have defined the clock settings.

Arrange Icons (Window Menu)

Select this command to arrange the iconized windows horizontally on the bottom of the Hardware Debugger main window.

Asynchronous Mode (Debug Menu)

Select this command to set the debug mode to asynchronous. Debug is available only if the cable selected is an XChecker cable and if the current file is a BIT file. In asynchronous mode, neither the Hardware Debugger nor the XChecker cable controls the clock for the target FPGA.

Figure 8-1 Asynchronous Mode Toolbar Button

After you select the asynchronous debug mode, the Settings submenu options are enabled on the Debug menu and the control panel is displayed on the main screen. Make sure you configure the XChecker cable as specified for the asynchronous mode.

Axes (View Menu)

Use this View → Change Color option to modify the color of the current waveform axes.

1. Click on **Axes** from the Change Color submenu of the View menu to display the Axes dialog box.
2. Select the appropriate color and click on the **OK** button to apply the change to the waveform.

Refer to the Change Color command for details.

Background (View Menu)

Use this View → Change Color option to modify the background color of the current waveform.

1. Click on **Background** from the Change Color submenu of the View menu to display the Background dialog box.
2. Select the appropriate color and click on the **OK** button to apply the change to the waveform window.

Refer to the Change Color command for details.

Cascade (Window Menu)

Select this command to display each new window on top of the preceding window in cascade fashion.

Change Color (View Menu)

The following options are available from this command:

- Signals: changes the signal colors.
- Axes: changes the axes colors.
- Background: changes the color of your background.
- Grid: changes the grid color.

When you select one of the above options, a Change Color dialog box is displayed for the waveform or screen element you chose. Figure 8-2 is the dialog box that appears when you selected a signal

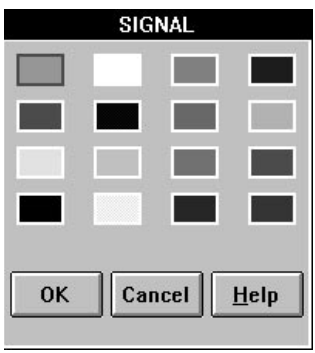


Figure 8-2 Change Color Dialog Box for Signals

Use the Change Color dialog box to change the color of your waveform elements. To restore the default colors, use the Reset Color to Default command.

Change Color Dialog Box Options

The Change Color dialog box includes the following options:

- OK button

  Click on this button to accept the settings defined on this dialog box and exit the dialog box. The software updates the waveform with the selected color.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Help button

  Click on this button to get context-sensitive help.

Change Radix (View Menu)

Select this command to display a dialog from which you can change the base or radix used by the Hardware Debugger to annotate your groups with readback values.

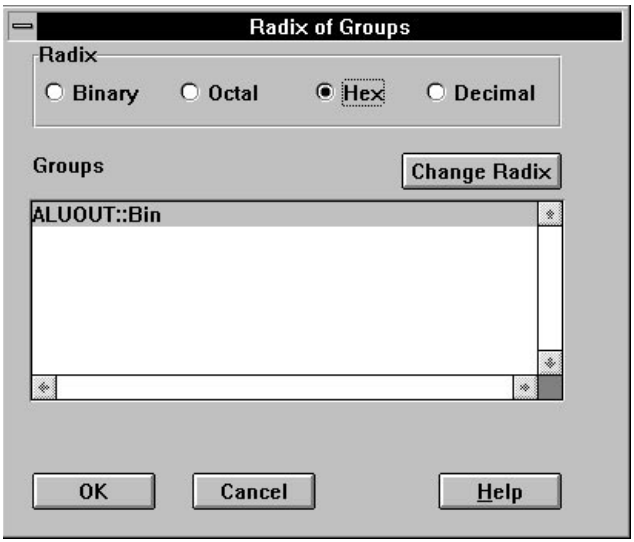


Figure 8-3 Radix of Groups Dialog Box

Use the Radix of Groups dialog box shown in Figure 8-3 to change the radix or base used by the Hardware Debugger to annotate your groups with readback values.

You can specify different radices for different signal groups provided you make the changes one at a time.

Radix of Groups Dialog Box Options

The Radix of Groups dialog box includes the following options:

- Radix box

  The radix dialog box shows the available radices. There are four possible radices: binary, octal, decimal, and hexadecimal. Select the radix you want to use for your signal groups.

- Groups list box

  This list box displays the groups that are present in the current waveform. Select one or more groups whose radix you wish to modify.

- Change Radix button

  This button is enabled when you select one or more signal groups in the Groups list box. Before clicking on this button, select the new radix you want to use for your groups.

- OK button

  Click on this button to accept the settings defined on this dialog box and exit the dialog box. The software updates the radix of the modified signal groups on the waveform.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Help button

  Click on this button to get context-sensitive help.

Clock (Debug Menu)

Clock... *Control Panel Button*

Select this Debug → Settings menu command to invoke the Clock Settings dialog box, which enables you to select the clock you will be using, **Internal Clock** or **External Clock**, by clicking on the appropriate radio button.

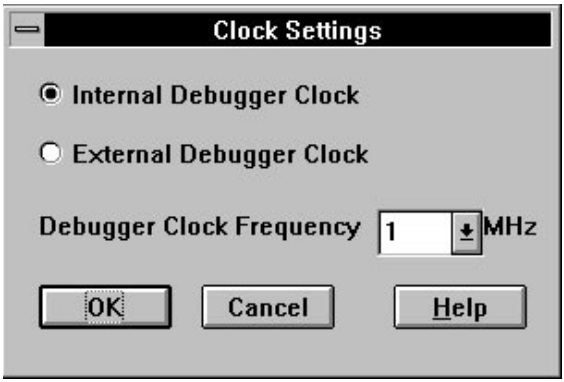


Figure 8-4 Clock Settings Dialog Box

Use the Clock Settings dialog box to define the clock options.

Clock Settings Dialog Box Options

The Clock Settings dialog box includes the following options:

- Internal Clock radio button

  If you select the internal clock, the target FPGA obtains the clock from the XChecker cable internal clock. In this case, you can also select the clock speed using the Hardware Debugger Clock Frequency box.

- External Clock radio button

  If you select the external clock, the target FPGA uses the system clock. In this case, you cannot set the clock speed. The list box for the Clock Frequency is disabled. When you click on the OK button, the clock settings are defined and are displayed on the Status Bar.

- Debugger Clock Frequency box

  You can set the frequency of the internal clock at 1, 3, 5, or 11 Mhz.

  For 3 Volt devices, the only allowable speed is 155 KHz; therefore, it is the only speed shown in the Frequency box.

- OK button

  Click on this button to accept the clock settings you have defined on this screen and exit the screen.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Help button

  Click on this button to get context-sensitive help.

Close (File Menu)

Select this command to close any active window. This command is active only if there is an active window.

Note: Before closing a window that contains readback data, you might want to save the contents of that window for future reference.

Communications (Cable Menu)

Select this command to invoke a dialog box on which you specify the Port Settings for downloading and debugging. Once defined, these settings are saved for the next program session.

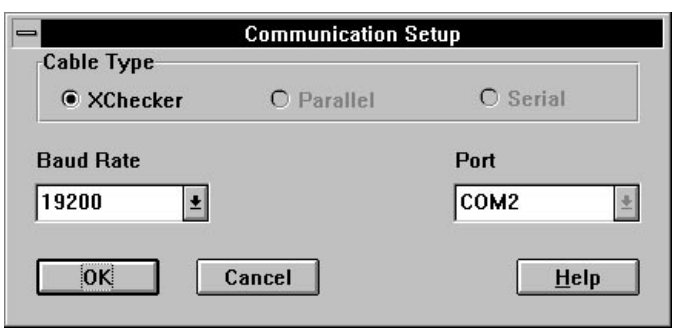


Figure 8-5 Communication Setup Dialog Box

From the Communications Setup dialog box, displayed in Figure 8-5, you can configure the following cable options: cable type, baud rate, and port.

Communication Setup Dialog Box Options

The Communication Setup dialog box includes the following options:

- Cable Type group box

  You can choose between the XChecker, parallel, and serial cables.

  - The XChecker cable allows you to download, verify, and debug.
  - The parallel and the serial cables allow you to download only.

- Baud Rate list box

  Specify a Communications Baud rate between the selected cable and the host system by selecting the desired baud rate from the Baud Rate pull-down list box.

- Port list box

  You can specify the port to be used for downloading and readback using the Port pull-down list box. This dialog box contains a list of valid ports.

- OK button

  Click on this button to accept the defined settings and exit this dialog box.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Help button

  Click on this button to get context-sensitive help.

Console (View Menu)

Use this option, located under the View menu, to open or close the Console window. By default, the Console is closed.

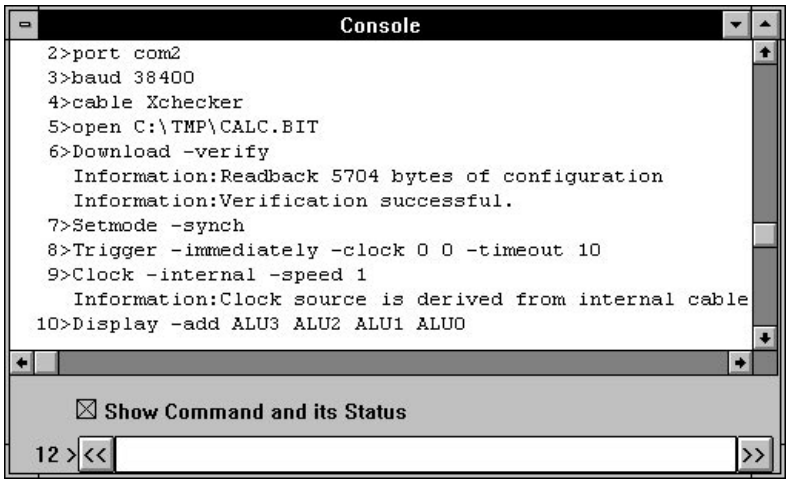


Figure 8-6 Console Window

The Console window, displayed in Figure 8-6, consists of a display and an option used to specify the status mode. Use the Console window to view the command history. You can also execute commands from the Console window command bar.

Console Window Options

The Console window includes the following options:

- Display Area

  The Console window is for read access only, that is you cannot edit the commands in the display. However, you can copy the commands and paste them in a macro window.

  Click on a command with the left mouse button to select the entire command line. Use the Shift key to select multiple consecutive lines when used with the mouse.

- Command Bar

  The command bar is located at the bottom of the Console window. You can enter Hardware Debugger commands using the appropriate syntax. Commands must also be entered in the right sequence, and you will be notified by a message if you enter

commands in the wrong order. Refer to the appendix, “Console Commands,” for command syntax information.

- Show Command and Its Status check box

  Select this option to display the commands in the window with the status information that was provided at the time the commands were executed.

  Deselect this option to display the commands only.

Note: To exit the Console window, select Console from the View menu.

Contents (Help Menu)

Select this command to display a dialog box that lists the contents of the Help facility. To select the command from the toolbar, point the mouse cursor at the toolbar Help button and double-click on the left mouse button.

Figure 8-7 Help Toolbar Button

To bring up context-sensitive help for a specific menu item, click once on the Help button, drag the Help question mark onto the menu item you wish to get help on, and click on that item. The Help page available for that item is displayed.

Control Panel (View Menu)

Select this command to display or hide the Control panel. This command is not available until you select a debugging mode, Synchronous Mode or Asynchronous Mode, from the Debug menu.

Control Panel for Synchronous Mode

In synchronous mode, the Control panel displays the following debugging options in the form of buttons that you can click on or text boxes that you can type directly into. Each button is also available as a menu option from the Debug menu.

To set up the readback options, use the buttons in the Readback Setup group box. See Table 8-1.

Table 8-1 Readback Setup

Control Panel Button	Readback Setup Group Box
Clock	Invokes the Clock Settings dialog box for you to specify the type and frequency of the clock.
Trigger	Invokes the Synchronous Trigger Settings dialog box for you to select the type of trigger and number of clock cycles.
Groups	Invokes the Signal Groups dialog box to enable you to define signal groups for debugging.
Display	Invokes the Display Signals dialog box to enable you to include signals, groups, and RAM Bits in the display list for debugging.
New Waveform	Displays a new graphical waveform window.

To read back a device, use the buttons in the Readback Control group box listed in Table 8-2.

Table 8-2 Readback Control

Control Panel Button	Readback Control Group Box
Snapshots (Text Box)	Allows you to enter the number of snapshots to generate and display in the debugging window.
Read	Reads back the target device and captures snapshots for the states of the signals selected for debugging.
Reset	Resets the flip-flops in the target device.

To control the clock, use the buttons in the Clock Control group box listed in Table 8-3.

Table 8-3 Clock Control

Control Panel Button	Clock Control Group Box
Clocks ... (Text Box)	Allows you to enter the number of clock cycles to apply to your device.
Apply	Applies the clock cycles specified in the Clocks... text box.
Stop	Interrupts the system clock being applied to the target device.
Resume	Reapplies the system clock to the target device.

Control Panel for Asynchronous Mode

In the asynchronous mode, the Control panel displays the following debugging options in the form of buttons that you can click on or text boxes that you can type directly into. Each button is also available as a menu option from the Debug menu.

To set up the readback options, use the buttons in the Readback Setup group box. See Table 8-4.

Table 8-4 Readback Setup

Control Panel Button	Readback Setup Group Box
Trigger	Invokes the Asynchronous Trigger Settings dialog box for you to select the type of trigger.
Groups	Invokes the Signal Groups dialog box to enable you to define signal groups for debugging.
Display	Invokes the Display Signals dialog box to enable you to include signals, groups, and RAM Bits in the display list for debugging.
New Waveform	Displays a new graphical waveform window.

To read back a device, use the buttons in the Readback Control group box listed in Table 8-5.

Table 8-5 Readback Control

Control Panel Button	Readback Control Group Box
Snapshots (Text Box)	Allows you to enter the number of snapshots to generate and display in the debugging window.
Read	Reads back the target device and captures snapshots for the states of the signals selected for debugging.
Reset	Resets the flip-flops in the target device.

Display Signals (Debug Menu)

Display... *Control Panel Button*

Select this Debug → Settings menu command to invoke the Display Signals dialog box, which enables you to select the signals, groups, and RAM bits you wish to view.

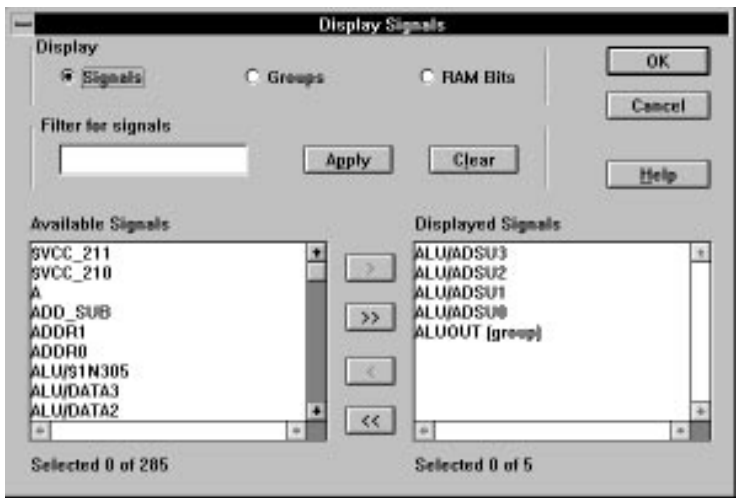


Figure 8-8 Display Signals Dialog Box

Use the Display Signals dialog box displayed in Figure 8-8 to specify which signals you wish to probe.

Display Signals Dialog Box Options

The Display Signals dialog box includes the following options:

- Display box

  Click on the Signals, Groups, or RAM Bits radio buttons to display the list of available signals, signal groups, or RAM bits.

- Filter for Signals box

  You can use the Filter for Signals box to globally define signals, signal groups, or RAM bits you wish to probe. For example, if you specify the letter A followed by a wildcard character, you will filter out all signal names that do not start with the letter A.

- Available Signals and Displayed Signals list boxes

  In the Signals list box, highlight one or more of the signals, signal groups, or RAM bits you wish to view. Then move the signals to the Displayed Signals list box using the right arrow (>) button. Use the left arrow (<) button to remove a signal from the list of displayed signals.

  Click on the double left-arrow (<<) button or the double right-arrow (>>) button to move all signals from one column to the other.

- OK button

  Click on this button to exit from the Display Signals dialog box.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Help button

  Click on this button to get context-sensitive help.

Download Design (Download Menu)

Select this command to download the current configuration data to a device or device chain. When you download a design, you program a device or daisy chain of devices, with the functions described by that design.

Figure 8-9 Download Design Toolbar Button

This command is active only if you have opened a design file.

1. Check the pin connections of your cable.
2. Power your board, if you have not already done so, and set the cable communications options.
3. Select the **Reset** option from the Cable menu to reset the internal logic of the cable and initiate a reprogram of the XChecker cable internal FPGA.
4. Select **Logic Level of Header Pins** from the Cable menu to verify the logic levels of the cable. Make changes to the pin configuration if necessary.
5. Open a design.
6. Select **Download Design** from the Download menu.

   The Download Status window in Figure 8-10 is displayed.

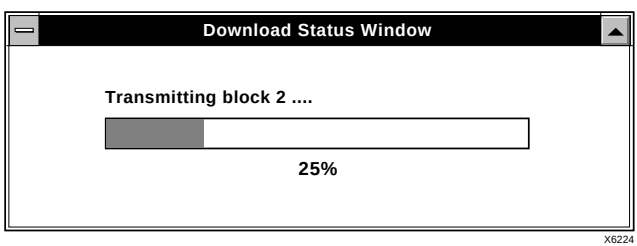


Figure 8-10 Download Status Window

Download and Verify (Download Menu)

Select this command to download a configuration bitstream and verify it immediately after. The output of the verification is displayed in a dialog box. This command is active only if you are using an XChecker cable and if the current file is a BIT file configured for readback.

Figure 8-11 Download and Verify Toolbar Button

1. Check the pin connections of your XChecker cable.
2. Power your board, if you have not already done so, and set the cable communications options.
3. Select the **Reset** option from the Cable menu to reset the internal logic of the cable and initiate a reprogram of the XChecker cable internal FPGA.
4. Select **Logic Level of Header Pins** from the Cable menu to verify the logic levels of the XChecker cable and make changes to the pin configuration if necessary.
5. Open a design.
6. Select **Download and Verify** from the Download menu.

Exit (File Menu)

Select this command to terminate the current Hardware Debugger session. If you have not saved your data, you are asked if the unsaved information should be saved before exiting.

Full (View Menu)

Select this command to display the entire waveform.

Figure 8-12 Zoom Full Toolbar Button

Grid (View Menu)

Use this View → Change Color menu option to modify the color of the grid of the current waveform.

1. Click on **Grid** from the Change Color submenu of the View menu to display the Grid dialog box.
2. Select the appropriate color and click on the **OK** button to apply the change to the waveform window.

Refer to the Change Color command for more details.

Logic Level of Header Pins (Cable Menu)

Select this command to display the logic levels of the cable pins and to check that the pins are connected properly.

The logic level of the following pins is provided for the XChecker cable:

```
RESET=1, INIT=1, D/P=0, PROG=1, CCLK=1, TRIG=0,
RT=1, RD=1.
```

Check the logic levels of your header pins before programming or reading back a device. Before downloading a file, the value of the D/P pin should be 0; after downloading the design, the value of the D/P pin should be 1.

Note: Ensure you do not connect the XChecker leads to signals of different voltage levels than required as you might damage the XChecker internal hardware.

Do not connect the control signals to XChecker before Vcc and Ground. The input/output pins of their internal FPGA should always be at a lower or equal potential than the rail voltage to avoid internal damage.

Move Signal (View Menu)

Select this command to move a signal from a location on the Waveform window to another location so as to modify the order in which the signal waveforms are displayed.

1. Select the signal in the waveform.
2. Select the **Move** command from the View menu.

   The cursor arrow icon turns into a large + icon to indicate that a move is pending.

3. Click on the signal waveform after which you wish to insert the selected signal.

You can only move one signal waveform at a time.

New Macro (File Menu)

Select this command to open a new macro window. A macro is a sequence of commands that you can execute much like a batch file.

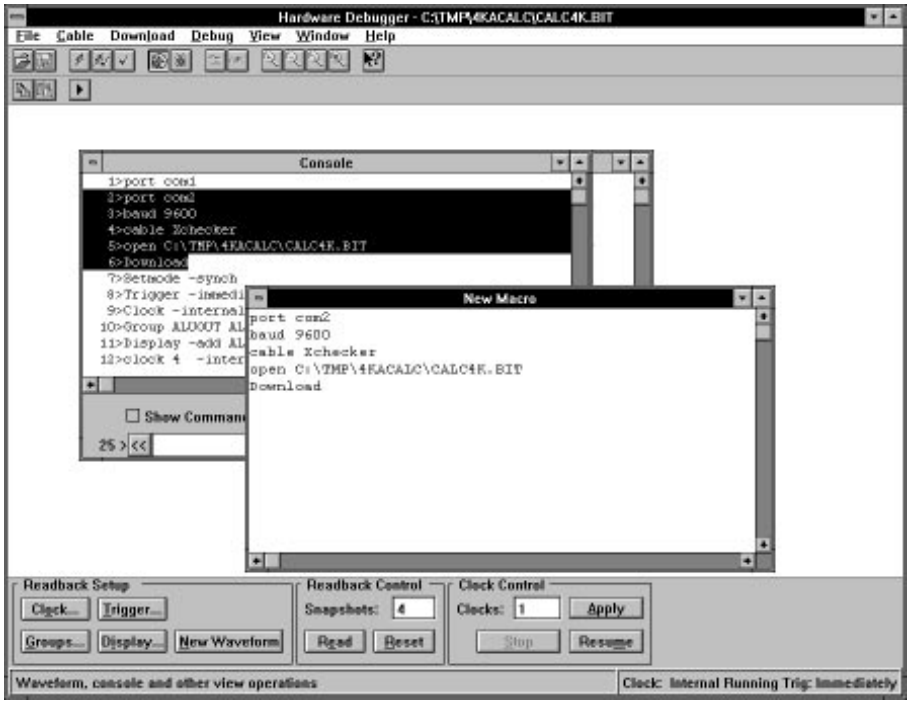


Figure 8-13 New Macro Window

Contrary to the Console window, a macro window includes an editable text area in which you can type the commands you want to use in the macro. The macro toolbar is displayed as well.

Display Area

You can select commands using the mouse by double-clicking the left mouse button to select a word. You can also select an entire block of text by pressing the right mouse button, dragging the mouse to the end of the region you wish to copy, then releasing the mouse button.

Macro Toolbar Buttons

- Select the Copy toolbar button, displayed in Figure 8-14, to copy the commands currently selected in the Console window to the clipboard. The button is available only if you have selected commands in the Console window or the macro window.

Figure 8-14 Copy Macro Toolbar Button

- Select the Paste button, displayed in Figure 8-15, to paste the contents of the clipboard at the insertion point in the macro text display.

Figure 8-15 Paste Macro Toolbar Button

- Select the Run Macro button, displayed in Figure 8-16, to run a macro.

Figure 8-16 Run Macro Toolbar Button

New Waveform (Debug Menu)

New Waveform *Control Panel Button*

Select this command to open a new graphical waveform window. If you select this option, the next time you click on Read FPGA, the

readback signal values will be displayed in the new waveform instead of the existing waveform.

Number of Clocks to Apply (Debug Menu)

Clocks *Control Panel Text Box*

Select this command from the Debug menu to open the Apply Clocks dialog box, shown in Figure 8-17, which enables you to specify the number of clocks to apply. When you select the Apply Clock command, the Hardware Debugger uses this clock number as the number of clocks to apply to the system.

By specifying the number of clocks, you are specifying the number of clock pulses to apply to the CLKO pin of the XChecker cable to the target system.

Figure 8-17 Apply Clocks Dialog Box

1. To invoke the Apply Clocks dialog box, select **Number of Clocks to Apply** from the Debug Settings menu.
2. Specify the number of clocks.
3. Click on **OK**.

Alternatively, you can directly type in the number you want in the Clocks... text box of the Control panel.

Number of Snapshots to Read (Debug Menu)

Snapshots... *Control Panel Text Box*

Select this command from the Debug menu to invoke the Snapshots Count dialog box, displayed in Figure 8-18, which enables you to specify the number of snapshots to be read. This number is used when you select the Read FPGA command. Alternatively, you can

type the number of desired snapshots in the Snapshots text box on the Control panel.

By specifying the number of snapshots, you are specifying the number of states you wish to display and probe.

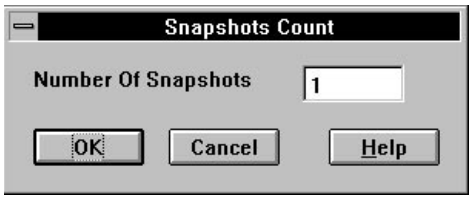


Figure 8-18 Snapshots Count Dialog Box

Use the Snapshots Count dialog box to specify the number of snapshots you want to generate and display in the current waveform. A snapshot is defined as the reading of the logical states of a device for one clock cycle.

Note: The maximum number of snapshots you can view on a single waveform is 450.

1. To invoke the Snapshots Count dialog box, select **Number of Snapshots to Read** from the Debug Settings menu.
2. Specify the number of snapshots.
3. Click on **OK**.

Alternatively, you can directly type in the number you want in the Snapshots text box of the Control panel.

Open Bitstream (File Menu)

Select this command to open a BIT file (.bit extension), which is a binary representation of your design, or a PROM file (.mcs, .tek, or .exo extensions). You can also open a Raw Bits file, called an RBT file (.rbt extension), which is the ASCII version of the BIT file. You must open the bitstream before downloading or debugging your design.

Figure 8-19 Open Bitstream Toolbar Button

Complete the Open Design File dialog box displayed in Figure 8-20 to specify which design file to open.

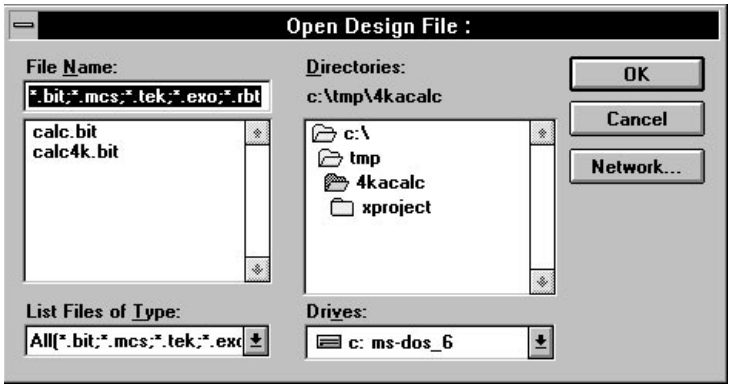


Figure 8-20 Open Design File Dialog Box

Open Design File Dialog Box Options

The Open Design File dialog box includes the following options:

- File Name box

  This box uses the file extension you specified in the List Files of Type box to filter the files in the current directory. Only the files that match the specified extension are displayed in the list box below it. You can use wildcards and letters to sort out various files. Alternatively, you can type in the filename and extension of the desired file in this box.

- Directories box

  This box displays the directories available for the drive currently selected. To change directories, click on the desired directory in the Directories box.

- List Files of Type box

  Use this box to specify which file extensions to use when searching for a file. Specify an extension by selecting the extension or group of extensions you want from the pull-down list box.

- Drives box

  Use this box to specify which drive to use. To select a drive, click on the Drives box down arrow. Choose the drive from the pull-down list box. When you select a drive, the software automatically updates the Directories box with that drive.

- OK button

  Click on this button after specifying the above options to open the design. The path and file name of your design are displayed in the title bar at the top of the screen.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Network button

  Click on this button to access the network drive. This option is only available if you are running Windows for Workgroups.

Note: If you opened a BIT file, you can download and debug your design. To verify, read back, and debug a BIT file, you must have a logic allocation file in the design directory.

If you are downloading a PROM file, you may program several FPGA devices at once. You cannot use PROM or RBT files for verification and debugging.

Open Macro (File Menu)

Select this command to bring up the Open Macro File dialog box from which you can specify which macro to open. The macro runs Hardware Debugger commands, which can range from a single step, such as setting the baud rate, to a full downloading and debugging session.

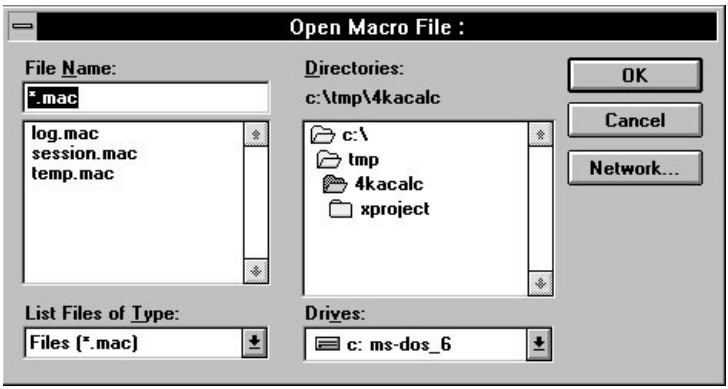


Figure 8-21 Open Macro File Dialog Box

Complete the Open Macro File dialog box displayed in Figure 8-21 to specify which macro file to open.

Open Macro File Dialog Box Options

The Open Macro dialog box includes the following options:

- File Name box

  This box displays the .mac file extension selected in the List Files of Type box. The list box, below it, displays existing files that have the specified extension.

  Type a file name in the File Name box. The .mac file extension is automatically appended when you click on the OK button.

- Directories box

  This box displays the directories available for the drive currently selected. To change directories, click on the desired directory in the Directories box.

- List Files of Type box

  This box displays the file extension .mac, which is used when you search for a file.

- Drives box

  Use this box to specify which drive to use. To select a drive, click on the Drives box down arrow. Choose the drive from the pull-down list box. When you select a drive, the software also updates the Directories box with that drive.

- OK button

  Click on this button after specifying the above options to open the selected file.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Network button

  Click on this button to access the network drive. This option is only available if you are running Windows for Workgroups.

Open Readback Data (File Menu)

Select this command to view previously saved readback data. When the file is first opened, the data is displayed graphically. To view the data textually, use the Text Data option of the View menu.

To open the file, specify the file name and press the OK button. A waveform is displayed as a waveform window within the Hardware Debugger main window.

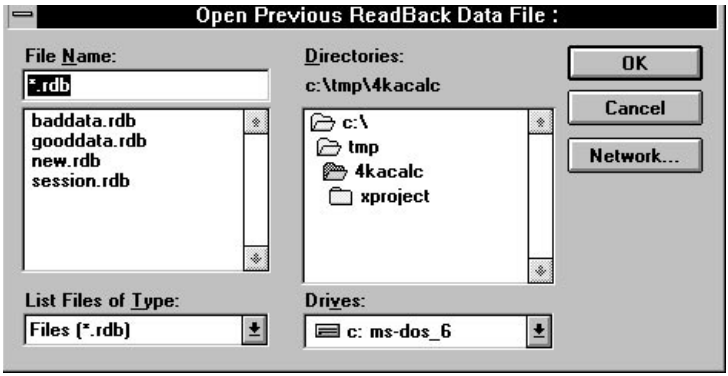


Figure 8-22 Open Previous Readback Data File Dialog Box

Complete the Open Previous Readback Data File dialog box displayed in Figure 8-22 to specify which readback data file to open.

Open Previous Readback Data File Dialog Box Options

The Open Previous Readback Data dialog box includes the following options:

- File Name box

  This box displays the .rdb file extension currently selected in the List Files of Type box. You may use this box to type the extension of the files you wish to search.

- Directories box

  This box displays the directories available for the drive currently selected. To change directories, click on the desired directory in the Directories box.

- List Files of Type box

  This box displays the file extension .rdb, which is used when you search for a file.

- Drives box

  Use this box to specify which drive to use. To select a drive, click on the Drives box down arrow. Choose the drive from the drop-

down list box. When you select a drive, the software also updates the Directories box with that drive.

- OK button

  Click on this button after specifying the above options to open the file. A waveform in the form of an MDI child window appears within the Hardware Debugger area.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Network button

  Click on this button to access the network drive. This option is only available if you are running Windows for Workgroups.

Print (File Menu)

Select this command to print the data from the selected waveform window. A message box appears immediately after you invoke the Print command. To stop printing, click on the Cancel button on the message box.

Print Setup (File Menu)

Select this command to invoke the Print Setup dialog box.

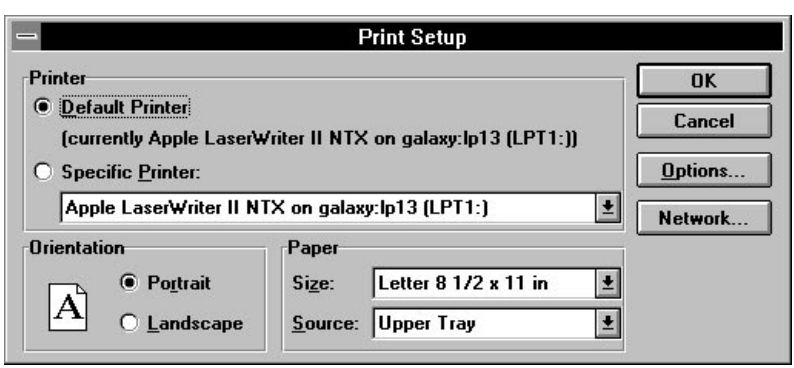


Figure 8-23 Print Setup Dialog Box

Use the Print Setup window displayed in Figure 8-23 to specify the printer options.

Print Setup Dialog Box Options

The Print Setup dialog box includes the following options:

- Printer group box

  Select this option to specify the printer you wish to use: Default Printer or a Specific Printer.

  If you select Specific Printer, use the pull-down list box to select one of the available printers.

- Orientation group box

  Select this option to specify the orientation of the printout: Portrait (vertical) or Landscape (horizontal).

- Paper group box

  Select this option to specify the dimensions of the document and the source from the pull-down list boxes.

- OK button

  Click on this button to accept the defined settings and exit this dialog box.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Options button

  Click on this button to access additional customization features.

- Network button

  Click on this button to access the network drive. This option is only available if you are running Windows for Workgroups.

Read FPGA (Debug Menu)

Read *Control Panel Button*

Select this command to read back the signals that you selected for viewing. The software generates a readback of all the configuration

data, extracts the signals you selected for viewing, and displays the signal values in the current waveform.

Note: The readback data stream is linked to the Active Waveform window only. Once the connection is closed, the waveform window will no longer be updated with new information. The maximum number of snapshots you can view in a single waveform is 450.

Refresh (View Menu)

Select this command to refresh and update the waveform with the latest signals.

Reset Color to Default (View Menu)

Select this command to use the default colors for the current waveform.

Reset (Cable Menu)

Select this command to reset the internal logic of the cable. After the cable is reinitialized and the baud rate is set, a dialog box confirms the cable type, port connection, and baud rate.

After a power glitch, you should first reset the cable to reconfigure the XChecker internal FPGA device and then reconfigure the target device.

Reset FPGA (Debug Menu)

`Reset` *Control Panel Button*

Select this command to reset the target FPGA internal flip-flops. If you have not set the Reset Before Readback option, you can use the Reset button to reset the target device's internal flip-flops.

Resume Clock (Debug Menu)

`Resume` *Control Panel Button*

Select this command to reapply the system clock after interrupting it. This option is available only after you have selected the synchronous debugging mode.

Run Macro (File Menu)

Select this command to run the commands in the selected macro window. This command is available only if a macro window is selected.

Figure 8-24 Run Macro Toolbar Button

Alternatively, to execute a macro, click on the Run Macro button located on the macro toolbar.

Save Macro/Save Macro As (File Menu)

Select the Save Macro command to save a new macro or an existing macro. If you are saving a new macro, the Save Macro As dialog box is displayed allowing you to specify the name under which you save the macro. If you save an existing macro that you modified, the Hardware Debugger saves the changes to the existing file without displaying a dialog box.

To save under a new name an existing macro displayed in the current macro window, use the Save Macro As command.

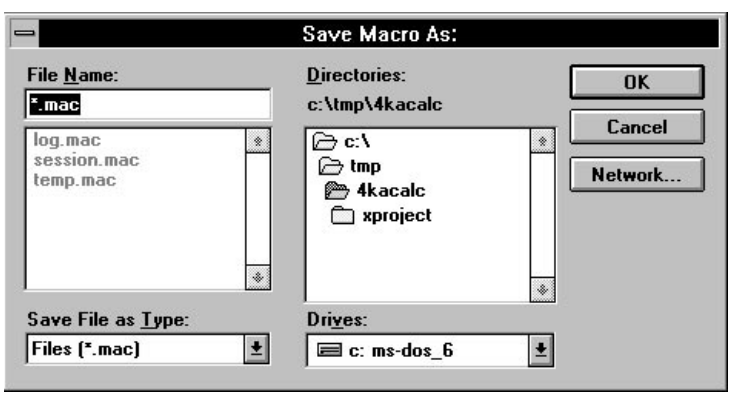


Figure 8-25 Save Macro As Dialog Box

Use the Save Macro As dialog box displayed in Figure 8-25 to save the selected macro window.

Save Macro As Dialog Box Options

The Save Macro As dialog box includes the following options:

- File Name box

  This box displays the .mac file extension selected in the List Files of Type box. The list box, below it, displays existing files that have the specified extension.

  Type a file name in the File Name box. The file extension is automatically appended when you click on the OK button.

- Directories box

  This box displays the directories available for the drive currently selected. To change directories, click on the desired directory in the Directories box.

- List Files of Type box

  This box displays the file extension .mac, which is used when you search for a file.

- Drives box

  Use this box to specify which drive to use. To select a drive, click on the Drives box down arrow. Choose the drive from the pull-down list box. When you select a drive, the software also updates the Directories box with that drive.

- OK button

  Click on this button after specifying the above options to save the data.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Network button

  Click on this button to access the network drive. This option is only available if you are running Windows for Workgroups.

Save Readback (File Menu)

Select this command to save the Readback data from the current waveform to the specified file.

Figure 8-26 Save Readback Toolbar Button

Depending on which window you are currently using, a waveform, a text window, the file extension appropriate for the window is displayed in the File Name box:

- RDB: This file contains all waveform-specific formats.
- TXT: This file contains the data in textual format.

Note: From the Hardware Debugger, you can re-open only waveforms saved in the graphical mode as RDB files, as explained in the section "Open Readback Data." You cannot open waveforms saved in the text mode as TXT files from the Hardware Debugger; therefore you cannot reuse waveforms saved in text format with the Hardware Debugger.

Complete the Save Readback Data As dialog box displayed in Figure 8-27 to save the debugging results displayed in the current waveform window.

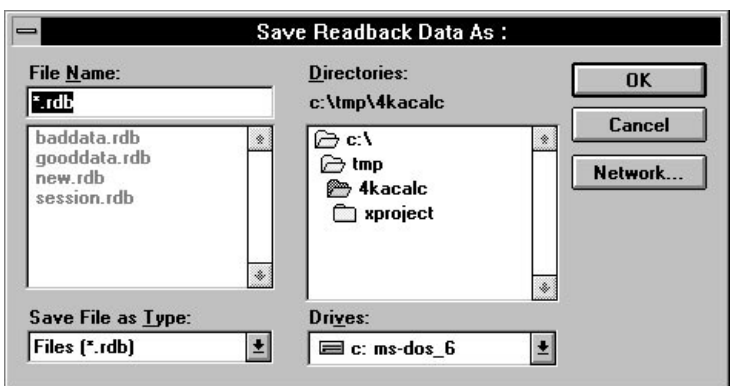


Figure 8-27 Save Readback Data As Dialog Box

Save Readback Data As Dialog Box Options

The Save Readback Data As dialog box includes the following options:

- File Name box

  This box displays the file extension currently selected in the List Files of Type box. The file extension corresponds to the type of Readback data you will save. The list box, below it, lists existing files that have the specified extension.

  Type a file name in the File Name box. The file extension is automatically appended when you click on the OK button.

Note: To save readback data for later use, view it graphically first, then save it as a graphical waveform with the .rdb extension.

- Directories box

  This box displays the directories available for the drive currently selected. To change directories, click on the desired directory in the Directories box.

- List Files of Type box

  This box displays the file extension appropriate for the window currently displayed: a waveform or a text window.

- Drives box

  Use this box to specify which drive to use. To select a drive, click on the Drives box down arrow. Choose the drive from the pull-down list box. When you select a drive, the software also updates the Directories box with that drive.

- OK button

  Click on this button after specifying the above options to save the data.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Network button

  Click on this button to access the network drive. This option is only available if you are running Windows for Workgroups.

Search for Help on (Help Menu)

Select this command to display the Search dialog box from which you can locate a particular Help topic.

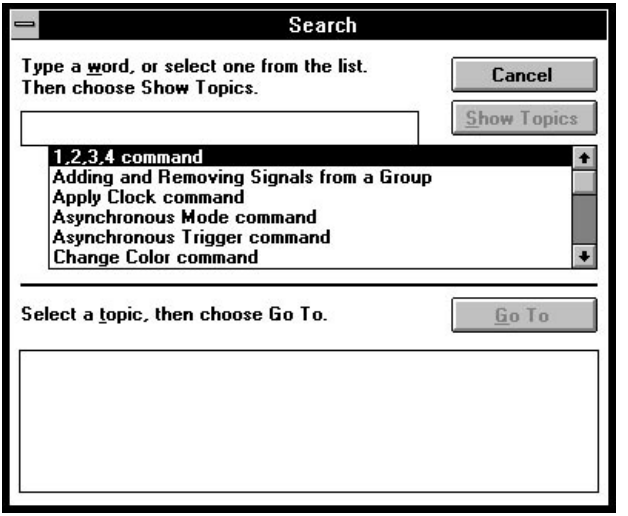


Figure 8-28 Search Dialog Box

Use the Search dialog box displayed in Figure 8-28 to locate a Help topic. You are prompted to enter a word in the box where the cursor is located. You may also select a topic from the list box.

Search Dialog Box Options

The Search dialog box includes the following options:

- Show Topics button

  Click on this button to display the related topics in the lower box of the Search dialog box.

- GoTo button

Click on this button to close the Search dialog box and open the desired topic.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

Self-Check (Cable Menu)

Select this command to perform a diagnostic of the XChecker cable. The tests ensure that data can be transmitted and received at the different baud rates and that the internal FPGA and RAM are operational.

1. Insert the test fixture, which is provided with the XChecker cable, onto the XChecker cable pins.

   The test fixture is a small printed circuit board with a keyed header connector that fits onto the XChecker cable pins.

2. XChecker draws power from your target system, not from the host system. To this effect, the test fixture has two connectors: a red V_{CC} power (+5 V) connector and a black Ground connector. Connect these signals to a 5 V DC power supply.

3. Start the Hardware Debugger software or click on **`Reset`** from the Cable menu to establish communication with the cable.

4. Select **Self-Check** from the Cable menu to perform a diagnostic of the XChecker cable.

   The command invokes a dialog box with the message "`Place Test Fixture on the cable and connect it to` V_{CC} `and Ground`."

5. Press **`OK`** to start the diagnostic. The output is displayed in a Status window.

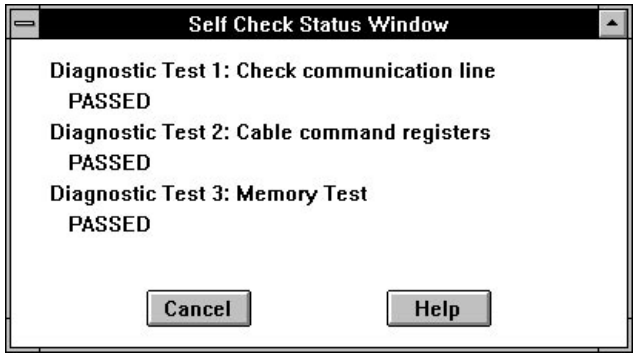


Figure 8-29 Self-Check Status Window

The Self-Check command performs a check of the XChecker cable hardware. When you run the Self-Check command, a status window is displayed.

You can follow the progress of the diagnostics routines from that window.

Self-Check Diagnostics

The diagnostics routines consist of software loops that verify the functions of the cable. The following tests are run:

- Communication Line: This test checks the RS-232 port on the host system at the different baud rates.
- Cable Command Registers: This test checks the operation of the XChecker cable header pins by verifying that the Hardware Debugger commands are functional.
- Memory Test: Checks the cable 1MB RAM that is used to store, download, and readback data.

Each time a test is completed, you are informed whether the test was successful or not. If the test is successful, the status window displays PASSED under the test name. If the test is unsuccessful, the status window displays FAILED and a description of the test failure under the test name.

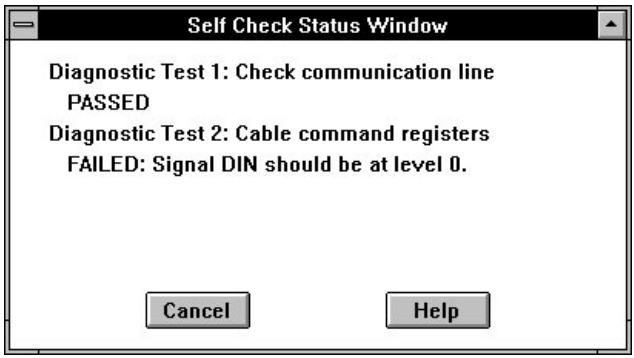


Figure 8-30 Failed Self-Check Test

Troubleshooting

- A failure in the Communication Line test indicates that the XChecker cable is not connected correctly or that the serial port of your host system is defective.
- A failure in the Command Registers test indicates that one or more XChecker pins are damaged.
- A failure in the Memory Test indicates that the cable's RAM is damaged.

If the Command Registers test or the Memory test fails, you should replace the XChecker cable.

Settings (Debug Menu)

The Settings submenu options are enabled only if you have selected a debugging mode, Synchronous Mode or Asynchronous Mode.

The following options are available if you select **Synchronous Mode** debugging:

- Clock
- Trigger
- Display Signals
- Signal Groups

- Number of Snapshots to Read
- Number of Clocks to Apply

The following options are available if you select **Asynchronous Mode** debugging:

- Trigger
- Display Signals
- Signal Groups

Signals (View Menu)

Use this View → Change Color menu option to change the color of your signals.

1. Select a signal on the waveform window.
2. Click on **Signals** from the Change Color submenu to display the Signal Color dialog.
3. Select the appropriate color and click on the **OK** button to apply the change to the waveform.

Refer to the Change Colors command for more details.

Signal Groups (Debug Menu)

Groups... *Control Panel Button*

Select this Debug → Settings menu command to invoke the Signal Groups dialog box, which contains a list of signal groups and a list of signals. The dialog box enables you to add or remove signals from a group. You can modify and delete groups that are not included in the Displayed Signals list of the Display Signals dialog box.

Signal groups are useful for probing nodes that are related.

For example, the signals a0, a1, a2, and a3 of a bus can be grouped as a signal address group.

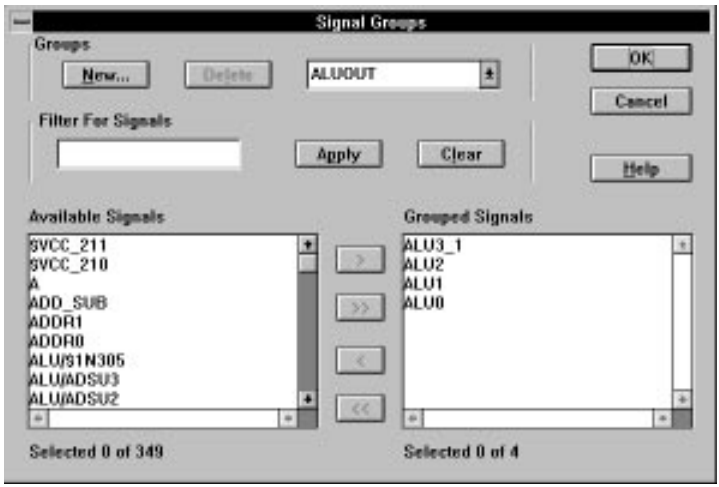


Figure 8-31 Signal Groups Dialog Box

Use the Signal Groups dialog box displayed in Figure 8-31 to define signal groups.

Signal Groups Dialog Box Options

The Signal Groups dialog box includes the following options:

- Groups box

  To create a new group, click on the New button to invoke the Group Name dialog box from which you can specify a new group name. Click on the Group Name OK button to exit that dialog after specifying the new group name.

  To delete a group name, click on the down arrow located to the right of the pull-down list box to display the list of defined groups. Select a group from the list. After you select a group, the Delete button is enabled. Click on the Delete button to remove the group from the list.

- Filter for Signals box

  You can use the Filter for Signals box to globally define signals you wish to include under the same group. For example, if you

specify the letter A followed by a wildcard character, you will filter out all signal names that do not start with the A character.

- Apply button

  To filter the list of signals for the specified signals, click on the Apply button.

- Clear button

  To display the complete list of signals, click on the Clear button.

- Available Signals and Grouped Signals list boxes

  The left column contains a list of selectable signals that can be included in the current group. The right column contains the list of signals included in the group.

  Select the signals you want to include in a group by highlighting one or more signals at a time and using the right arrow (>) button to move the highlighted signals to the right hand side column. To remove a signal from the list of included signals, select the signal from the right hand column and use the left arrow (<) button.

  Click on the double right-arrow (>>) button or the double left-arrow (<<) button to move all signals from one column to the other.

- OK button

  Click on this button to save the group and exit from the Signal Group dialog box.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Help button

  Click on this button to get context-sensitive help.

Status Bar (View Menu)

Select this command to display or hide the Status Bar, located at the bottom of the Hardware Debugger window. The status bar provides information about the commands being processed. It also provides a description of menu commands or the toolbar icons that are in focus.

Stop Clock (Debug Menu)

Stop *Control Panel Button*

Select this command to interrupt the system clock being applied to the target device through the CLKO pin of the XChecker cable.

This option is available only after you have selected the synchronous debugging mode.

Synchronous Mode (Debug Menu)

Select this command to set the debug mode to synchronous debugging. Debugging is available only if the cable selected is an XChecker cable and if the current file is a BIT file. In synchronous mode, you control the FPGA system clock through the Hardware Debugger and the XChecker cable.

Figure 8-32 Synchronous Mode Toolbar Button

After you select a debug mode, the Settings submenu options are enabled on the Debug menu and the control panel is displayed on the main screen. Make sure you configure the XChecker cable as specified for the synchronous mode.

Text Data (View Menu)

Select this command to display the readback data textually rather than graphically.

Figure 8-33 Text Data Toolbar Button

In the text mode, the waveform window displays a textual representation of the device states you specified for debugging.The text window displayed in Figure 8-34 is a tabular display of the readback information.

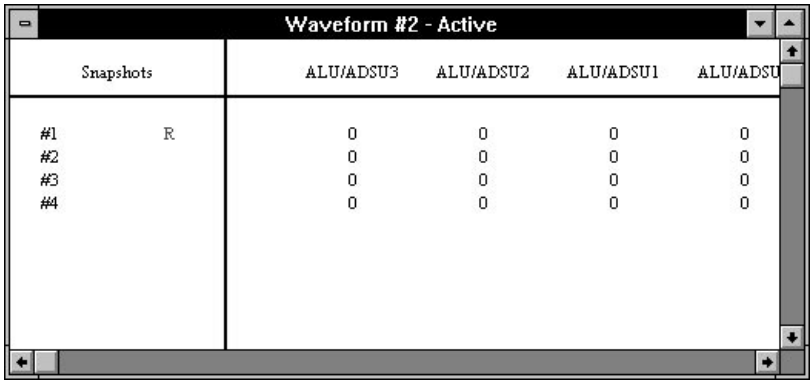


Figure 8-34 Text Data Waveform Window

Window Options

- Signals and Signal Groups

  The display area shows the readback values of signals and signal groups.

  The left-most column lists the snapshots in sequential order. The signals and groups are listed horizontally at the top of the table. The readback value of each signal is listed under the signal and in the row corresponding to the snapshot for which the value was captured.

  A reset is indicated by a capital R to the left of the vertical axis.

- Vertical Axis

  The vertical axis on the waveform separates the snapshots from the signal names and values.

Tile (Window Menu)

Select this command to arrange open windows next to one another within the main window so that all windows are legible. The window that is in focus is titled Active Window and will process the commands you invoke.

Toolbar (View Menu)

Select this command to display or hide the toolbar located at the top of the Hardware Debugger window. The toolbar displays several icons, which you can use to specify commands directly. The toolbar offers an alternative to the menu commands.

Figure 8-35 Open Design File Toolbar Button

The Open Design File button opens a configuration data file.

Figure 8-36 Save Readback Toolbar Button

The Save Readback button saves the contents of the current waveform window.

Figure 8-37 Download Toolbar Button

The Download Bitstream button downloads the configuration data to your target device or chain of devices.

Figure 8-38 Download and Verify Toolbar Button

The Download Design and Verify button downloads the design file to your target device and verifies that the design was downloaded correctly.

Figure 8-39 Verify Toolbar Button

The Verify Bitstream button verifies that the design was downloaded correctly.

Figure 8-40 Synchronous Debug Toolbar Button

The Synchronous Debug button enables the synchronous debug mode and displays the control panel.

Figure 8-41 Asynchronous Debug Toolbar Button

The Asynchronous Debug button enables the asynchronous debug mode and displays the control panel.

Figure 8-42 Waveform Display Toolbar Button

The Waveform Display button displays the readback data graphically.

Figure 8-43 Text Display Toolbar Button

The Text Display button displays the readback data in text format.

Figure 8-44 Zoom In Toolbar Button

The Zoom In button increases the size of your waveform

Figure 8-45 Zoom Out Toolbar Button

The Zoom Out button decreases the size of your waveform.

Figure 8-46 Zoom Region Toolbar Button

The Zoom Region button enables you to delimit a region to be displayed.

Figure 8-47 Zoom Full Toolbar Button

The Zoom Full button displays the entire waveform.

Figure 8-48 Help Toolbar Button

The Help button invokes help for a specific menu item or object on your screen.

1. Click once on this button to display the help cursor.
2. Position the help cursor on the object you need help on.

3. Press the left mouse button to display the corresponding Help page.

Trigger — Async (Debug Menu)

Trigger... *Control Panel Button*

Select this Debug → Settings menu command to invoke the Asynchronous Trigger Settings dialog box, which enables you to select the type of trigger you wish to initiate the readback. The trigger is an external signal that tells XChecker to start the readback operation.

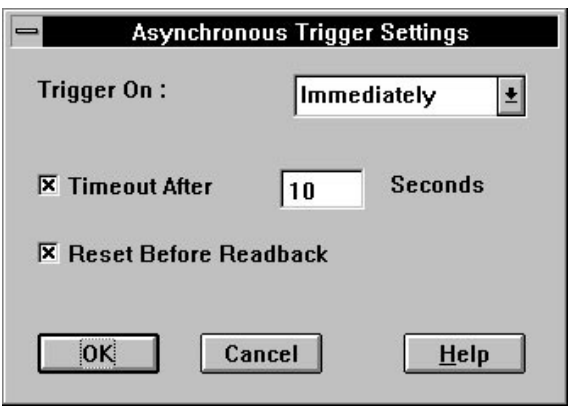


Figure 8-49 Asynchronous Trigger Settings Dialog Box

Complete the Asynchronous Trigger Settings dialog box displayed in Figure 8-49 to specify the type of trigger to use for debugging.

Asynchronous Trigger Settings Dialog Box Options

The Asynchronous Trigger Settings dialog box includes the following options:

- Trigger On list box

  You may wish to initiate the readback **Immediately** (default), that is when the Read FPGA command is invoked, or you may choose to use an **External Pin** or the **Enter Key** to initiate the readback.

The External Trigger is an auto-trigger that is activated when the TRIG pin of the XChecker cable goes from Low (0) to High (1).

- Timeout After box

  Use this option to specify the cutoff time for a trigger to be detected. If the trigger is not received within the specified number of seconds, the readback is canceled. Valid values range from 0 to 99.

- Reset Before Readback box

  Use this option to clear the device flip-flops. By default, this field is deselected.

- OK button

  Click on this button to accept the settings you have defined on this screen and exit the dialog box.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Help button

  Click on this button to get context-sensitive help.

Trigger — Sync (Debug Menu)

Trigger.. *Control Panel Button*

Select this Debug → Settings menu command to invoke the Synchronous Trigger Settings dialog box, which enables you to select the type of trigger you wish to initiate the readback. The trigger is a signal that tells XChecker to start the readback operation.

The Number of Clock Cycles option enables you to define specific clock patterns. When used with the Number of Snapshots to Read command, you can use these clock patterns to gather the device states you want.

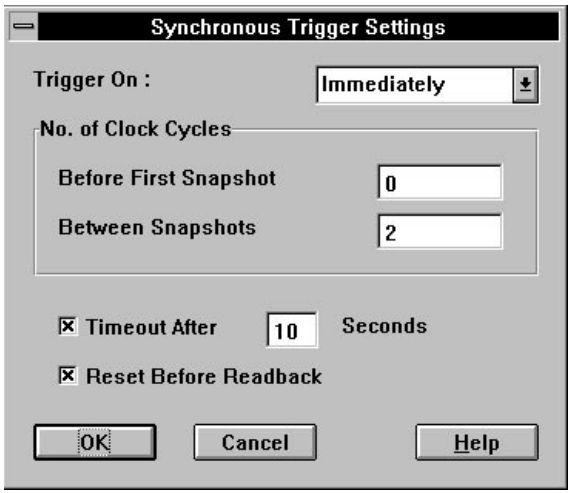


Figure 8-50 Synchronous Trigger Settings Dialog Box

Complete the Synchronous Trigger Settings dialog box displayed in Figure 8-50 to specify the trigger type and clock cycles to use for debugging

Synchronous Trigger Settings Dialog Box Options

The Synchronous Trigger Settings dialog box includes the following options:

- Trigger On list box

  You may wish to initiate the readback **Immediately** (default), that is when the ReadFPGA command is invoked, or you may choose to use an **External Pin** or the **Enter Key** to initiate the readback.

  The External Trigger is an auto-trigger that is activated when the TRIG pin of the XChecker cable goes from Low (0) to High (1).

- Number of Clock Cycles group box

  Use this option together with the Number of Snapshots to Read option to specify the states of the bitstream you wish to display and probe.

This option allows you to specify the number of clock cycles to be applied before the first snapshot and between snapshots. You use this option to advance the device a specific number of states before the first snapshot and to determine how many clocks are applied to the target FPGA between snapshots.

- Timeout After box

  Use this option to specify the cutoff time for a trigger to be detected. If the trigger is not received within the specified number of seconds, the Readback is canceled. Valid values range from 0 to 99.

- Reset Before Readback box

  Use this option to clear the device flip-flops. By default, this field is deselected.

- OK button

  Click on this button to accept the defined settings and exit the dialog box.

- Cancel button

  Click on this button to exit the current dialog box without saving the settings.

- Help button

  Click on this button to get context-sensitive help.

Tutorial (Help Menu)

Select this command to invoke the interactive tutorial.

Verify Bitstream (Download Menu)

Select this command to initiate a readback of the downloaded design and compare the readback data to the original bitstream that was downloaded. The output of the verification is displayed in a dialog

box. This command is active only if you are using an XChecker cable and if the current file is a BIT file configured for verification.

Figure 8-51 Verify Toolbar Button

1. Power your board, if you have not already done so.
2. Check the pin connections of your XChecker cable.
3. Select the **Reset** option of the Cable menu to reset the internal logic of the cable and initiate a reprogram of the XChecker cable internal FPGA.
4. Select the **Logic Level of Header Pins** option of the Cable menu to verify the logic levels of the XChecker cable and make changes to the pin configuration if necessary.
5. Select the **Verify Bitstream** option of the Download menu.

Vertical Grid (View Menu)

Select this command to display or hide the vertical grid located on the Waveform window.

Waveform (View Menu)

Select this command to display the readback signal values graphically rather than textually.

Figure 8-52 Waveform Toolbar Button

The waveform window shown in Figure 8-53 displays a graphical representation of the device states you specified for debugging. To display the waveform textually, select the Text Data option from the View menu.

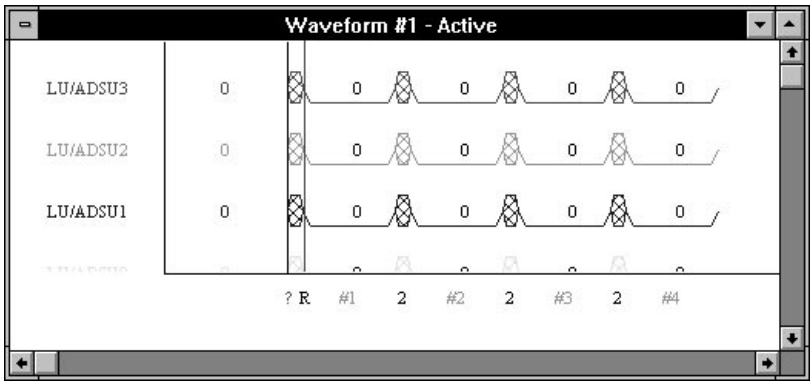


Figure 8-53 Waveform Window

Window Options

The Waveform window includes the following options:

- Signals and Signal Groups

  The display area shows waveforms for the readback values of signals and signal groups.

  - Each *Signal* is displayed as a semi-hexagon within which the readback value of the signal is displayed.
  - Each *Signal Group* is displayed as a complete hexagon. Each hexagon contains the concatenated value of all the signals included in the group at the time of a snapshot.
  - *Readback values* are separated by hexagonal cross-hatched patterns.

- Vertical Axes

  The two left-most vertical axes on the waveform separate the signal names from their values.

  - The signal and group names column, left of the first axis, shows the last eight characters of the signal names that you are viewing.
  - The current data column, left of the second vertical axis, shows the signal values during the last readback operation.

A red axis indicates a reset, and a capital R is displayed at the point where the reset axis intersects the horizontal axis.

- Horizontal Axis

  The horizontal axis marks the snapshots (#1, #2, #3...) and the number of clock cycles that take place before and in-between consecutive snapshots.

  - A capital R indicates a reset.
  - The first number in a readback series represents the clock number applied before the first snapshot is taken. It is part of the reset axis.
  - A question mark (?) indicates a free-running clock.

- Grid

  The horizontal axis uses equispaced marks to indicate snapshots, regardless of the number of clock cycles that might occur in-between them. At each marking for a snapshot, a pair of vertical lines representing the time period of the readback event is displayed. These pairs of vertical lines constitute the *grid*. To turn on the vertical grid that separates snapshots from one another, select Vertical Grid from the View menu.

Zoom In (View Menu)

Select this command to zoom in on the current waveform.

Figure 8-54 Zoom In Toolbar Button

Zoom Out (View Menu)

Select this command to zoom out on the current waveform.

Figure 8-55 Zoom Out Toolbar Button

Zoom to Region (View Menu)

Select this command to delimit a particular region to display. This command operates when a graphical waveform window is the active window.

Figure 8-56 Zoom to Region Toolbar Button

1. Click on the Zoom to Region button.
2. Using the mouse to delimit the area you wish to zoom on, position the mouse cursor at the beginning of the area you wish to select.
3. Press down the left mouse button and pan right with the mouse.

   A colored rectangular shape indicates the progress of the selection process.
4. Release the mouse when you have selected the area you wish to zoom into.

Glossary of Terms

This chapter is a list of terms commonly used in the Hardware Debugger program.

Definitions

The following terms are defined to help you understand the Hardware Debugger terminology.

aliases

Aliases, or signal groups, are useful for probing specific groups of nodes.

asynchronous debugging

A debugging mode in which you are using a free-running clock.

BIT file

A synonym for a configuration bitstream file.

bitstream (BIT file)

A data stream, also called BIT file, that contains location information of logic on a device, that is, the placement of CLBs, IOBs, TBUFs, pins, and routing elements. The bitstream also includes empty placeholders that are filled with the logical states sent by the device during a readback. Only the memory elements, such as flip-flops, RAMs, and CLB outputs, are mapped to these placeholders as their

contents are likely to change from one state to another. When downloaded to a device, a bitstream programs the device.

A bitstream file has a .bit extension.

CCLK pin

The pin of the configuration cable that connects the configuration clock to the device.

CLKI pin

XChecker clock input pin. This pin connects the system clock to the XChecker cable. This clock must be from 120 kHz to 10 MHz. Connecting the system clock to the CLKI pin allows XChecker to synchronize with the target system.

CLKO pin

XChecker clock output pin. This pin connects to the destination of the target system clock. If the clock source is external, the clock signal is connected to the CLKI pin; if it is internal, the clock signal is generated by the XChecker cable electronics.

console log

A record of the commands that you invoked during a Hardware Debugger session.

control panel

An area of the Hardware Debugger that consists of buttons and fields for debugging purposes. The control panel is displayed after you select a debugging mode in the Hardware Debugger. The control panel commands offer an alternative to the Debug menu commands.

debugging

The process of reading back or probing the states of a device previously configured with a design to ensure that the device is behaving normally.

DIN pin

The Data In pin of the configuration cable that connects to the DIN pin of your target device. The DIN pin loads the bitstream data to the target FPGA in serial mode.

D/P pin (XC3000)

The dual function Done/Program pin of your configuration cable. The pin connects to the D/P pin on your target device. As an input, D/P is used to initiate a device reconfiguration. As an output, D/P signals the end of configuration.

DONE pin (XC4000)

Dual function pin of your configuration cable. This pin connects to the DONE pin of your target FPGA. As an input, the DONE pin can be configured to delay the global logic initialization or the enabling of outputs. As an output, it indicates the completion of the configuration process.

downloading

Configuring or programming a device by sending bitstream data in serial or parallel mode.

external clock

The clock that XChecker uses from outside in synchronous mode. To use an external clock, connect the system clock to the XChecker cable using the CLKI pin and the XChecker clock to the FPGA device using the CLKO pin.

GND pin

Ground (0 volt) pin of the configuration cable. This pin connects to the Ground pin of a power supply.

group

A combination of signals that have a common output. In the case of a counter, for example, the different signals that produce the actual counter values can be grouped under the same name and share a common representation.

INIT pin

Initialization pin on your configuration cable. This pin is connected to the INIT pin of your target device indicating when a device is ready to receive configuration data after power up.

internal clock

The clock internal to the XChecker cable. This clock is generated in the XChecker cable electronics and is output on the CLKO pin. It is available in synchronous mode and is controlled from the Hardware Debugger.

(.ll) file

The logic allocation file, which indicates the bitstream position of storage elements, such as latches, flip-flops, and IOB inputs and outputs. The Hardware Debugger uses this file to locate signal values inside a readback bitstream. This file is created during the implementation process if readback is enabled in the Configuration Template.

main screen

The background against which the windows are displayed in the Hardware Debugger. The current window is titled Active window.

menu bar

The area located at the top of the Hardware Debugger window. It includes the File, Cable, Download, Debug, View, Window, and Help menus. Refer to the chapter titled “Command Reference” for information on the menu commands.

number of clock cycles

The value of each clock cycle, which represents the number of clocks that have occurred in-between snapshots. This number is the number of clocks between snapshots that you specified for the synchronous debugging mode. Each clock number is displayed at the point where the vertical snapshot line intersects the horizontal axis.

probing

The process of examining the states of an FPGA device.

PROG pin

The Program pin of your configuration cable that provides a reprogram pulse to XC4000 devices when connected to the PROG pin of the device.

PROM file

A PROM file is the file output by the PROM File Formatter, which can be used to program devices. The PROM File Formatter supports the following PROM file formats: MCS (Intel MCS-86), EXO (Motorola EXORMacs), and TEK (Tektronix hexadecimal).

RBT file

A raw BIT format file, the ASCII version of the BIT file.

readback

The process of reading the logic downloaded to an FPGA device. There are two types of readbacks:

- A readback of logic usually accompanied by a comparison check to verify that the design was downloaded in its entirety.
- A readback of the states stored in the device memory elements to ensure that the device is behaving as expected.

RD pin

The readback data pin of the XChecker cable. This pin connects to the device RDATA pin. When connected, the pin reads data from the programmed target device.

RST pin

The Reset pin of the XChecker cable that connects to the RST pin of the target device. This pin can be driven Low after configuration to reset the target FPGA internal latches and flip-flops.

RT pin

The readback trigger pin of the XChecker cable. This pin connects to the device RTRIG pin. When the XChecker cable drives this pin High, the pin initiates a readback on the target FPGA.

snapshot

An individual device state that was captured at a particular time that you can view in a waveform. A snapshot is also called readback.

Each snapshot *n*, is delimited by a pair of vertical dotted lines on the waveform provided you turned on the Grid option from the View menu.The number *n* that figures below the horizontal axis is the snapshot number of the readback event.

states

The values stored in the memory elements of a device (flip-flops, RAMs, CLB outputs, and IOBs) that represent the state of that device for a particular readback. To each state there corresponds a specific set of logical values.

status bar

The field located at the bottom of the Hardware Debugger window. It provides information about the commands that you are about to select or that are being processed.

synchronous debug

A debug mode in which you use the XChecker cable to have full control of the clock.

TCK pin

In the Hardware Debugger, an XChecker cable pin reserved for future use.

TDI pin

In the Hardware Debugger, an XChecker cable pin reserved for future use.

TMS pin

In the Hardware Debugger, an XChecker cable pin reserved for future use.

toolbar

A field located under the menu bar at the top of your screen. It contains a series of button that you click on to execute some of the most commonly used commands. These buttons constitute an alternative to the menu commands.

TRIG pin

The external trigger pin of the XChecker cable. This pin is connected to an external signal used as a trigger. A Low to High transition on this pin signals the XChecker cable to initiate a readback.

trigger

A signal that tells the Hardware Debugger to start the readback operation.

V_{CC} pin

Power (5 volt) pin of the XChecker cable. This pin connects to the power pin of a 5 volt power supply.

verification

The process of reading back the configuration data and comparing it to the original downloaded design to ensure that all of the design was received by the device.

waveform

The graphical representation of one or more readbacks. Usually, you select a set of signals and a set of readbacks for display. Each readback represents a particular state of the memory elements of the device.

CALC Tutorial

This chapter demonstrates how to use the Hardware Debugger to download, verify, and debug a single design using a Xilinx demonstration board as your target device.

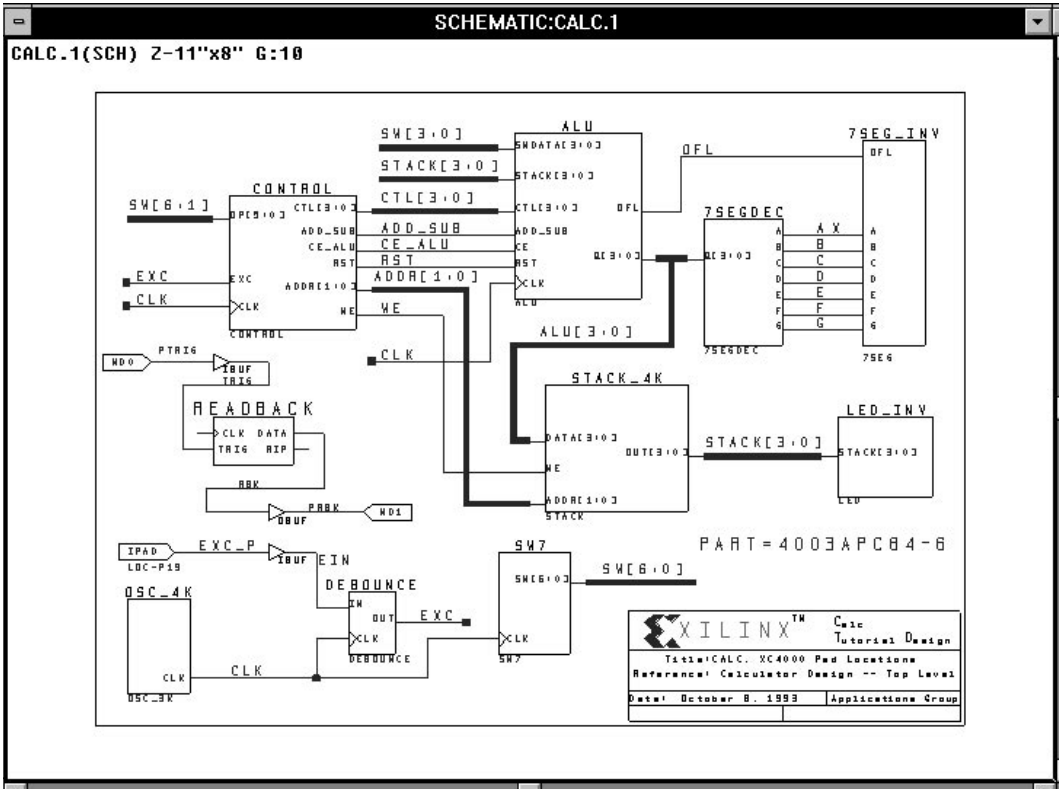


Figure 10-1 Calc Schematic

Although information is provided for all three demonstration board types — XC3000, XC4000, and FPGA demonstration boards — the tutorial uses an XC4000A design and is therefore targeted at the FPGA demonstration board. The tutorial design is called Calc and is located in the `/soln_4ka` directory. Figure 10-1 presents a complete view of the Calc schematic. For an XC3000A design, use the Calc design located in the `/soln_3ka` directory.

In addition to the demonstration board, you also need the XChecker cable. Although other cables are available from Xilinx, the XChecker cable is the only cable that has readback capabilities.

Note: This tutorial assumes that you are familiar with your schematic entry tool and the Xilinx Design Manager. It also assumes that you have implemented your design. If you are not familiar with these tools or have not implemented your design, complete the tutorials for those tools prior to proceeding.

Testing the Design Using a Demonstration Board

There are three Xilinx demonstration boards in common use. Which board you have depends on what software you purchased and when you bought it. Your tutorial design should be targeted to a device on the board that you have available.

- The FPGA demonstration board includes both an XC3000 family socket and an XC4000 family socket, containing an XC3020APC68 device and an XC4003APC84 device, respectively. This tutorial only uses one of the two parts; which part you use is up to you.
- Another board, referred to as the XC3000 Demonstration Board, contains a single XC3020PC68 device.
- The third board, referred to as the XC4000 Demonstration Board, contains a single XC4003PC84 or XC4003APC84 device.

Preparing the Design for Readback

Locate the Calc tutorial for the XC4003A device and edit it as explained in this section to enable readback. This step is not necessary if you intend only to download your design or if you are using an XC3000 design.

1. Include the READBACK symbol in your Calc schematic using a

schematic editor. Figure 10-1 shows the complete schematic with the Readback symbol.

2. Connect an IBUF and an OBUF primitive to the TRIG and DATA pins of the READBACK symbol.

   Figure 10-2 shows a detailed view of the READBACK symbol and its connections.

3. Use the MD0 and MD1 primitives if you want to use the FPGA dedicated RT and RD pins for readback. If you plan to use other readback pins, use the IPAD and OPAD symbols instead.

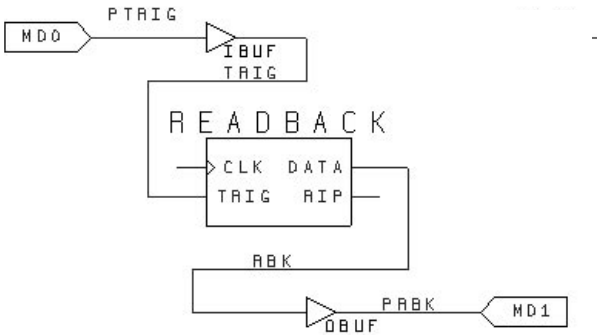


Figure 10-2 Readback Symbol Connections

Generating a Bitstream

The next step generates a bitstream for the Calc design from the Design Manager. To generate a bitstream, you must open the implemented Calc design from the Design Manager.

Note: If you do not have a project for the Calc design, use the Design Manager to create a project for that design. If a project already exists for the Calc design but is not implemented, run the Translate command from the Design menu to read in the changes made in the schematic and to create a new version reflecting the updated schematic, then implement the design as explained in the *Design Manager/Flow Engine Reference/User Guide*.

Set the Configuration Bitstream options as explained in this section to generate a configuration file that you can use for programming, verifying, and debugging XC4000 designs. You need to enable a pull-

up resistor for the DONE pin, which is used for device configuration. You also need to enable readback for the device.

Note: To generate configuration data for XC3000 devices, refer to the "Design Preparation" chapter in this manual.

1. From the Design Manager, select the routed design icon, shown in Figure 10-3.
2. Click the right mouse button and select **`Implement`** from the pop-up menu to access the Design Implementation Options dialog box.

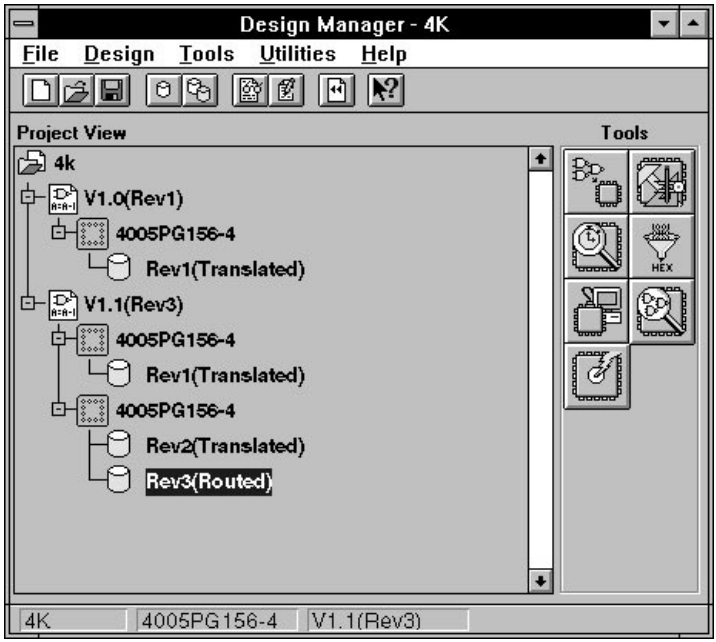


Figure 10-3 Design Manager Configuration Dialog Box (XC4000)

3. Click in the check box next to **`Produce Configuration Data`** in the Optional Targets group box to set Configuration Data as the target.
4. Click on the **`Edit Template`** button corresponding to the **`Configuration Program Options`** template.

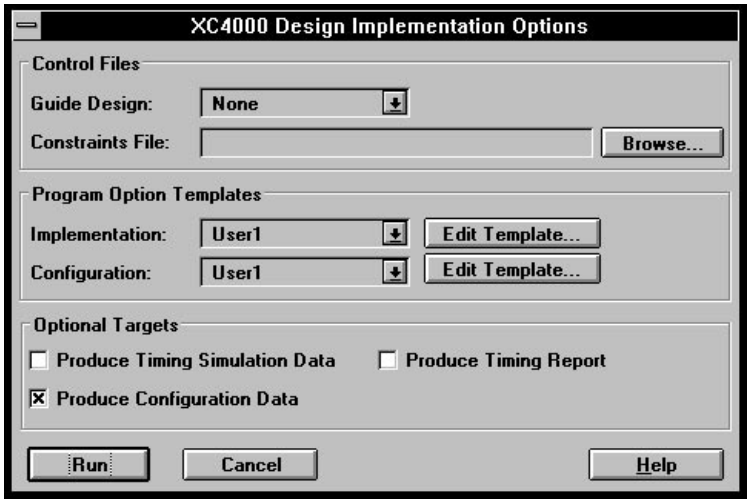


Figure 10-4 Design Manager Configuration

The Configuration Template dialog box is displayed as shown in Figure 10-5.

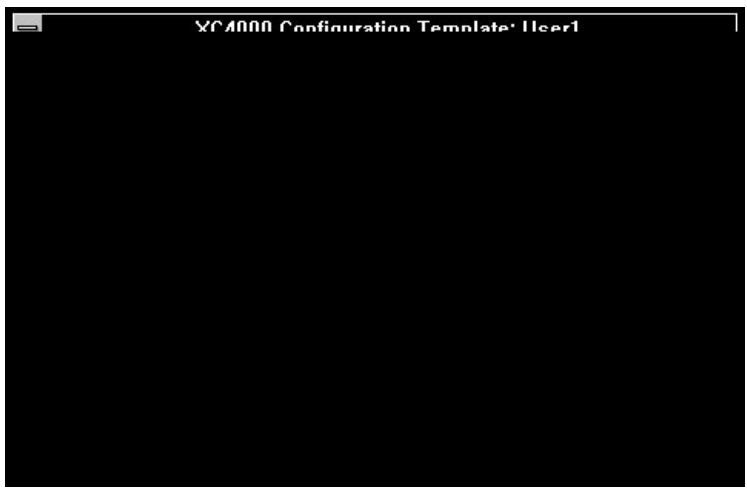


Figure 10-5 Design Manager Configuration Data Screen

5. Click on **Pull-Up** next to the DONE pin in the Configuration Pins box to enable a pull-up resistor for the DONE pin.
6. Select **Perform CRC During Configuration** to perform a CRC check of your bitstream during configuration.
7. Select **Produce ASCII Configuration File** to create a raw bits text (RBT) file, which is an ASCII representation of your configuration bitstream.
8. Select the **Readback** tab to access the Readback screen.

   The Readback screen is displayed as shown in Figure 10-6.
9. Select the **CCLK** readback clock.
10. Select **Capture CLB and IOB Outputs When TRIG Goes Active**. This option generates the logic allocation file.
11. Select **Abort Readback When TRIG Goes Inactive**.

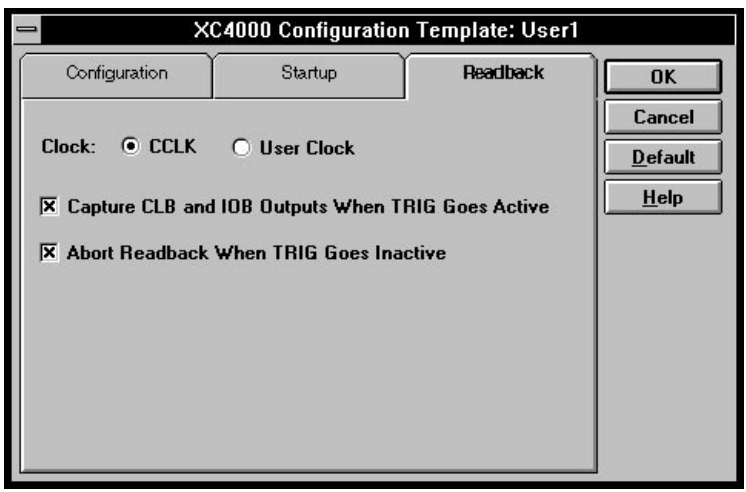


Figure 10-6 Design Manager Readback Data Screen

Note: For more information about configuration options, read the Configuration chapter in the *Design Manager/Flow Engine Reference/User Guide*.

12. Click on **OK** to return to the Implementation dialog box.

13. Select **Run** to compile the design and produce the configuration data.

Connecting the Cable

To load the configuration bitstream to the demonstration board, you need one of the three available hardware cables: the XChecker cable, the parallel cable, and the serial cable. All three cables work with any of the Xilinx demonstration boards; however, the XChecker cable is the only cable that supports verification and debugging.

Before initiating the physical downloading of the design into the FPGA on a Xilinx demonstration board, you must hook up the board correctly to your PC.

You must also connect several control and power pins between the board and the cable. The bundles of leads supplied with the cables are labeled to help you connect the board to the cable.

Finally, you must connect a pair of power and ground pins to a regulated 5 volt power supply to provide power to the board and cable.

1. Plug one end of the cable into the back of your machine.

   If you are using a parallel cable, attach the cable to a parallel port. If you are using a serial cable or the XChecker cable, connect that cable to a serial port.

2. Connect the other end of the cable to your demonstration board. Connections from the cable to the demonstration boards for downloading are shown in Table 10-1 for XC3000 and XC4000 device families.

   For the FPGA demonstration board, use the leftmost column of pins, labeled J1, which is missing the pin in the third position.

   The “XC3000 Design Demonstration Board”, “XC4000 Design Demonstration Board,” and “FPGA Demonstration Board” chapters of the *Hardware & Peripherals User Guide* discuss in detail the various demonstration boards and how to hook them up.

3. Connect the RT and RD pins, which are used for verification. Refer to Table 10-2 for pin location information.

4. To perform an asynchronous debugging, which is the purpose of this tutorial, connect the RT, RD, and TRIG pins but leave the CLKI and CLKO pins unconnected.

Note: For synchronous debugging, you would connect the RT, RD, and CLKO pins, and optionally, the TRIG and CLKI pins. See the "Connecting Your Cable" chapter for details.

5. Ensure that the power supply is connected to the demonstration board *and is turned on.*

   The power connections for the demonstration boards are shown in Table 10-3.

Table 10-1 Cable Connections (Downloading)

Cable Label	XC3000 and XC4000 Boards	FPGA Board	
		XC3000A	XC4000A
V_{CC}	J1-1	J1-1	J2-1
Gnd	J1-2	J1-3	J2-3
No Connection	J1-3	J1-5	J2-5
CCLK	J1-4	J1-7	J2-7
D/P	J1-5	J1-9	J2-9
DIN	J1-6	J1-11	J2-11
XChecker and Serial Download Cables for XC4000 Boards Only:			
PROG	J1-7 (XC4000)	NA	J2-13
XChecker Cable Only:			
INIT	J1-8 (XC4000)	NA	J2-15
RST	J1-9 (XC4000)	NA	J2-17

Table 10-2 Cable Connections (Verification and Debugging)

XChecker Cable Label	XC3000 Board	XC4000 Board	FPGA Board	
			XC3000A	XC4000A
CCLK	J1-4	J1-4	J1-7	J2-7
RT	25 (M0)	J3-1*	J1-19	J2-19
RD	26 (M1)	J3-2*	J1-21	J2-21
TRIG	user-assigned	J3-3	J1-23	J2-23
CLKI	user-assigned	J3-8	J1-25	J2-25
CLKO	user-assigned	J3-9	J1-27	J2-27

* If you do not use the MD0 and MD1 pads in your READBACK symbol shown in Figure 10-2, connect the XChecker RT and RD signals to the appropriate jumpers near the FPGA on the board. The RT pin connects to the pin that you assigned to the TRIG pin on the READBACK symbol while the RD pin connects to the pin that you assigned to the DATA pin. For information on which pins to use for verification or debugging, refer to the "Connecting Your Cable" chapter.

Table 10-3 Demonstration Board Power Connections

XC3000 Board		XC4000 Board		FPGA Board	
J3-1	+ 5 volts	J2-1	+5 volts	J9-1	+5 volts
J3-2	Gnd	J2-2	Gnd	J9-2	Gnd

XC3000 Demonstration Board

If you have an XC3000 demonstration board that has been modified for use with a serial PROM, such as an XC1736D or an XC1765D, be sure the board is not configured for use with that PROM. The modified board has sockets in which you can fit the serial PROMs. It also contains a four-position DIP switch with a power switch and three switches controlling the programming mode. If this DIP switch is present and you have a serial PROM on the board, make sure that the switches are set for slave mode download. Refer to the "XC3000 Design Demonstration Board" chapter of the *Hardware & Peripherals User Guide* for more information.

XC4000 Demonstration Board

Make sure the XC4000 demonstration board is set up for slave mode configuration. The configuration mode is controlled by the SW4 bank of switches. The switches should be set as shown in Table 10-4.

Table 10-4 SW4 Switch Settings for XC4000 Configuration

Switch	Label	Setting
SW4-1	PWR	Off (unless using battery)
SW4-2	MPE	Off
SW4-3	SPE	Off
SW4-4	M0	Off
SW4-5	M1	Off
SW4-6	M2	Off
SW4-7	RST	On
SW4-8	No Label	Don't Care

FPGA (XC3000A/XC4000A) Demonstration Board

Make sure the FPGA demonstration board is set up for slave mode configuration. The configuration mode for the XC3000A family part is controlled by the SW1 bank of switches. The configuration mode for the XC4000A family part is controlled by the SW2 bank of switches. Set the switches for the device you are using as shown in Table 10-5 for XC3000A devices or Table 10-6 for XC4000A devices.

Table 10-5 SW1 Switch Settings for XC3000A Configuration

Switch	Label	Setting
SW1-1	INP	Don't Care
SW1-2	MPE	Off
SW1-3	SPE	Off
SW1-4	M0	On
SW1-5	M1	On
SW1-6	M2	On
SW1-7	MCLK	Off
SW1-8	COUT	Off

Note: Besides setting the switches for the XC3000A device, you must ensure that SW2-8 (INIT) of the XC4000A device is off.

Table 10-6 SW2 Switch Settings for XC4000A Configuration

Switch	Label	Setting
SW2-1	PWR	Don't Care
SW2-2	MPE	Off
SW2-3	SPE	Off
SW2-4	M0	On
SW2-5	M1	On
SW2-6	M2	On
SW2-7	RST	Off
SW2-8	No Label	Off

Note: Besides setting the switches for the XC4000A device, you must toggle off SW1-7 (MCLK) and SW1-8 (DOUT) for the XC3000A.

Downloading and Verifying the Bitstream

Once the cable is connected to your PC, you are ready to download the bitstream. If you are using an XChecker cable, you can verify the design also.

1. Set all the input switches High to select the NOP (no operation) command:
   - On the FPGA board, set the SW3 switches High to the On position.
   - On the XC4000 board, set the SW5 switches High.
   - On the XC3000 board, set the SW1 switches High.
2. From the Design Manager, select the Hardware Debugger icon.

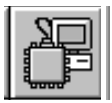


Figure 10-7 Hardware Debugger Icon

   You can use the software with any of the three hardware cables.

   The Communications dialog box is opened for you to specify the cable name, the port name, and the baud rate.

3. Select the correct port from the Port pull-down menu and the appropriate baud rate from the Baud Rate pull-down menu.

   Once you have used the cable and set the correct port, that information is saved in a file called *design*.ini in your design directory, so you do not have to specify it each time.

Note: If you invoke the Hardware Debugger from within the Design Manager after selecting the desired revision in the Project View, the configuration file is already loaded in the Hardware Debugger. If you did not set the project, complete steps 4 and 5, following, to load a bitstream file.

4. Select a file to open by selecting **`Open Bitstream`** from the File menu.
5. Select the input file name: **`CALC.BIT`** from the Open Design File box.

6. If you are using the serial Download cable to program an XC4000 family part, press the PROG button on the demonstration board. This step is not necessary if you are using the XChecker cable, the parallel cable, or if you are using the Download cable to program an XC3000 family part.
7. Select **Logic Level of Pins** from the Cable menu to check the status of the Done pin.
8. Select **Download** from the Download menu.

   If you are using an XChecker cable, select **Download and Verify** if you want to verify the design too. You must also connect the RT and RD pins for readback to be available.

   If the FPGA is successfully programmed, the following message appears:

   ```
   DONE signal went high.
   ```

   If the DONE signal does not go High, check the connections between the cable and the demonstration board, power the board off and on, and try downloading again. You must also ensure that you specified the correct device.

   If the Hardware Debugger informs you in a message that the current design does not include the READBACK block connected, check your schematic to ensure that the READBACK symbol is connected as shown in Figure 10-2.

Note: The Download cable has limited functionality when used with XC4000 family parts and may report that DONE went High even if you do not press the PROG button as in step 6, above. In this case the part is not reprogrammed. Download the bitstream again, this time pressing the PROG button prior to configuration. Cycling the power off and on before beginning the download has the same effect.

After downloading the design, the software initiates a design verification if you chose the Download and Verify option. The output of the design verification is displayed in a dialog box.

Number of bytes verified: 5703 bytes

Number displayed on 7-segment display: **. 0**

Note: The number of verified bytes is device specific.

Testing the Design

After programming your device with the logic functions of your design, you can test your design by checking the states of the device.

Each demonstration board has a row of eight rocker switches that provide input to the design (SW1 on the XC3000 board, SW3 on the FPGA board, and SW5 on the XC4000 board).

In the Calc design, the SW7 schematic assigns pin names to flip-flop inputs. Each pin corresponds to an input switch of the demonstration board. Table 10-7 lists the schematic labels of SW7 and the demo board switches to which the labels correspond. The interpretation of the pattern you use to set the switches depends on the pin assignment you used in the schematic. The different patterns generate different operation codes. Table 10-8 shows the operation codes assigned by the SW7 schematic to the bank of input switches of your demonstration board.

Each time you set the switches for a logical operation, you generate a value, which is stored in your flip-flops. When you toggle the Execute switch (switch 1), the value is displayed on the 7-segment display.

As described in the “Design Description“section towards the beginning of this tutorial, the Calc design is essentially a 4-bit processor with a stack. There are three types of inputs that you must supply using the input switches on your board: an Opcode, Data, and an Execute command.

Table 10-7 Schematic Labels vs. Demo Board Switches

Calc Design Signal	Demo Board Switch
SW0_P	SW*x*-8
SW1_P	SW*x*-7
SW2_P	SW*x*-6
SW3_P	SW*x*-5
SW4_P	SW*x*-4
SW5_P	SW*x*-3
SW6_P	SW*x*-2
EXC_P	SW*x*-1

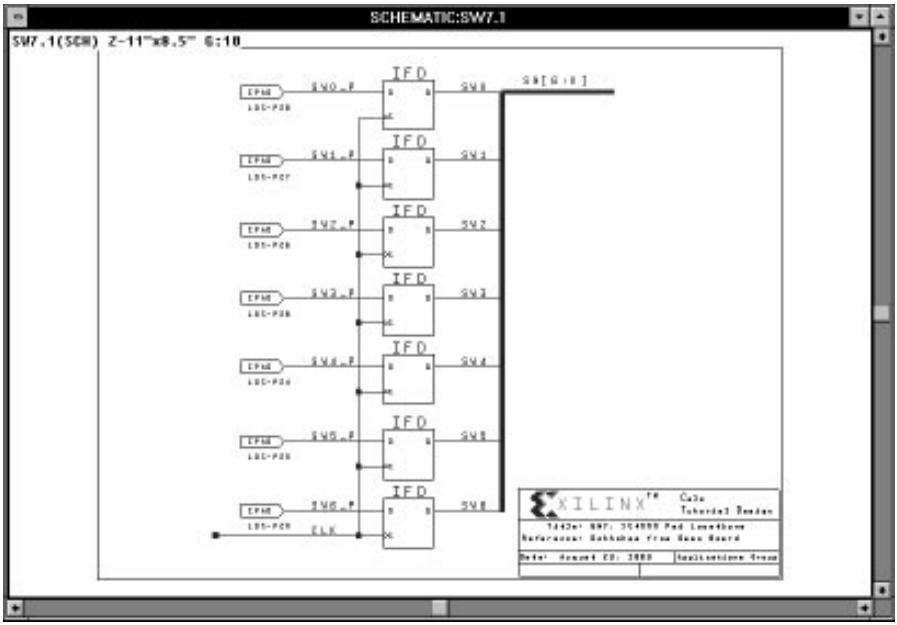


Figure 10-8 Module SW7

Use the demonstration board input switches to enter the opcodes and the data.

- The leftmost switch, labeled 1, is the Execute (EXC) command, which is activated by toggling the switch off and on.
- The next three switches (labeled 2, 3, and 4) select the opcode. Opcode encoding is shown in Table 10-8.
- Use the rightmost four switches (labeled 5, 6, 7, and 8) to input data.

When the extended instruction set is selected with opcode 111, these switches provide additional bits of opcode.

Note: The rocker switches on the XC3000 demonstration board are on when down, off when up. Use the 0 and 1 labels on the board as your guide.

Table 10-8 Operation Codes (Opcodes)

Operations	Demo Board Switches							Operations Between Switches and Register
	2	3	4	5	6	7	8	
	3-Digit Opcodes			Data				
ADD	0	0	0	*DATA*				ADD between switches and register
AND	0	0	1	*DATA*				AND between switches and register
OR	0	1	0	*DATA*				OR between switches and register
XOR	0	1	1	*DATA*				XOR between switches and register
SUB	1	0	0	*DATA*				SUB switch value from register
CLEAR	1	0	1	X	X	X	X	CLEAR register
LOAD	1	1	0	*DATA*				LOAD register

Operations	Demo Board Switches							Operations Between Stack and Register
	2	3	4	5	6	7	8	
	6-Digit Opcodes						Don't Care	
ADD	1	1	1	0	0	0	X	ADD between stack and register
AND	1	1	1	0	0	1	X	AND between stack and register
OR	1	1	1	0	1	0	X	OR between stack and register
XOR	1	1	1	0	1	1	X	XOR between stack and register
SUB	1	1	1	1	0	0	X	SUB stack value from register
PUSH	1	1	1	1	0	1	X	PUSH register value to stack
POP	1	1	1	1	1	0	X	POP stack value to register
NOP	1	1	1	1	1	1	X	NOP

Understanding Opcodes

Your Calc design includes a user-defined module, SW7, which consists of input pads, output pads, and flip-flops. The input pads are assigned pin locations. These pins are associated with the programming switches on your board. The flip-flops store the values displayed by the switches. The output pads display the values on your 7-segment display.

Operating the Board 7-Segment Display

To perform an operation, follow these guidelines:

1. Use the rightmost "nibble," (switches 5 to 8) to set the data.

   You can represent values ranging from 0 to (2^4–1). On is a one; Off is a zero.

2. Look up the correct opcode for the operation you want to perform and set the three opcode switches to the correct value.

3. Toggle the leftmost Execute (EXC) switch on.

   If the switch is already on, switch it off, wait a moment, and then return it to the On position.

   The contents of the ALU register are displayed in hexadecimal on the 7-segment display. The top value in the stack is displayed in binary on the bank of LEDs.

Opcode Examples

The following examples show you how to set the switches using operation codes to enter and process data arithmetically. For example, after loading data, you can complete the following operations: addition, subtraction, and boolean operations.

1. Verify that the initial contents of both ALU and stack are all zeros. The decimal display indicates 0, and the LED bar is off.

2. Now put a 1 on the DATA switches and load the switch value to the ALU register. The opcode for the load operation is 110.

   - Set the seven rightmost switches to **`110-0001`**.
   - Toggle the leftmost switch (switch 1) off and on to execute the command.

The decimal display shows the contents of the ALU register, which is now 1. The stack is still empty.

3. Add 13 to the ALU register. The opcode for the add operation is 000.

   - Set the seven rightmost switches to **`000-1101`**.
   - Toggle the leftmost switch (switch 1) off and on to execute the command.

     The 7-segment display shows the contents of the ALU register, which is now E. The stack is still empty.

4. Push the register value onto the stack. The opcode is 111, which is the extended opcode. The data must be set to 101*x*, where the *x* is a don't-care.

   - Set the seven rightmost switches to **`111-1011`**.
   - Toggle the leftmost switch off and on to execute the command.

     The decimal display still shows E. The stack value is also E, so the LED bar shows 1110 in the right hand side nibble.

5. Perform an XOR operation between the switch value and the register. The opcode is 011. Set the four data switches to ones.

   - Set the seven rightmost switches to **`011-1111`**.
   - Toggle the leftmost switch off and on to execute the command.

     The decimal display changes to 1. The stack value on the LED display is still E (hex), or 1110 (binary).

6. Pop the value from the stack. The opcode is 111, which is the extended opcode. The data must be set to 110*x*, where the *x* is a don't-care.

   - Set the seven rightmost switches to **`111-1101`**.
   - Toggle the leftmost switch off and on to execute the command.

     The decimal display changes to E. The stack value returns to 0.

7. Clear the ALU register. The opcode is 101. The data is ignored.
   - Set the seven rightmost switches to **101-1101**.
   - Toggle the leftmost switch off and on to execute the command.

     The decimal display changes to 0. The stack value remains at 0.
8. Try any other commands that you wish.

Debugging the Device

To display the states output by your device in a waveform instead of on your board digital display, use the debugging feature of the Hardware Debugger. To debug your design, you must set the debugging mode, set the Trigger type, and include signals and signal groups in your display list.

Setting the Debugging Mode

To debug the Calc design, you must use the asynchronous mode because the Hardware Debugger is not set up to control the design clock.

To set the debugging mode, follow these steps:

1. Select the **Asynchronous Debug Mode** from the Debug menu.

   The Control panel is displayed.
2. Select the **Trigger** option from the Control panel to display the Asynchronous Trigger dialog box and configure the trigger options as follows:
   - Select **Immediately** from the Trigger On pull-down menu.
   - Select the **Timeout After** box and set its value to **10** seconds.

Specifying Signal Groups

You now need to specify which signals to view. Follow these steps to add signals to the list of signals to display.

1. Select the **Groups** button on the Control panel to display the Signal Groups dialog box from which you can group signals into a bus for easy viewing.

Figure 10-9 Control Panel Buttons

The Signal Groups dialog box is displayed as shown in Figure 10-10.

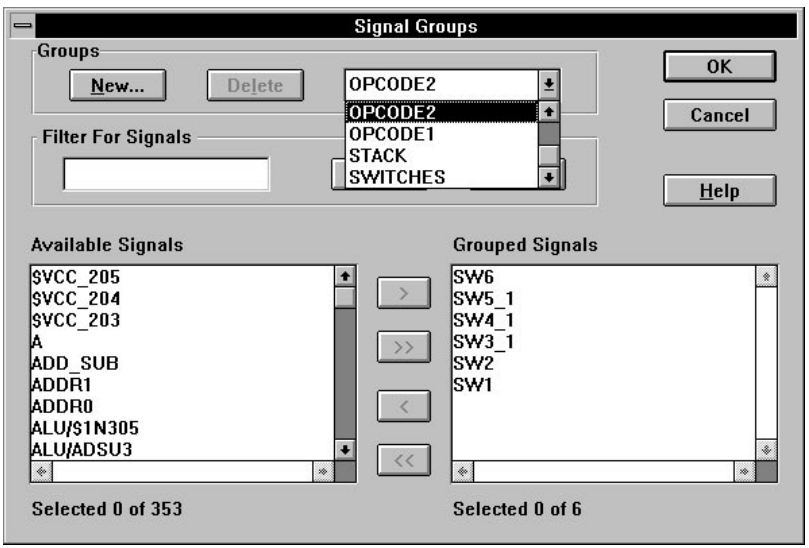


Figure 10-10 Signals Groups Dialog Box

2. To create a new group, click on **New** in the Groups group box. The Group Name box is displayed, as shown in Figure 10-11.

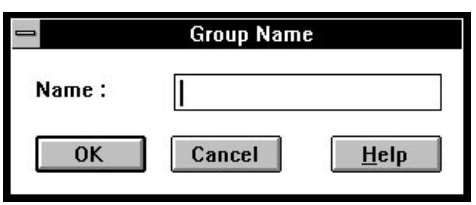


Figure 10-11 Group Name Dialog Box

3. Type the name of the bus you are creating in the Group Name box.
4. After entering a group name, click on the **OK** button of the Group Name box.

   The new group name appears in the Groups text box of the Signal Groups dialog box.
5. Select the signals to be grouped from the list of Available Signals and add them to the list of Grouped Signals. The top signal in the Grouped Signals list is the MSB of the bus.

   You can globally define the signals to display in the selection list-box by typing the first characters of the signals followed by a wildcard character (*) in the Filter for Signals box. Then click on the Apply button.

Note: If split nets appear in the list of signals, select the *signals*_1 nets. Signals that have an "_#" appended to the end of the name represent nets which have been split by the Place and Route program. Selecting any one of the split nets yields the same results during a readback.

6. Use the data in Table 10-9 to create groups that reflect the state of the switches. Each time you create a group, it is added to the pull-down menu in the Groups group box, as shown in Figure 10-10.

Table 10-9 Signal Groups

Group Name	Grouped Signals	Group Representation
SWITCHES	SW6, 5, 4, 3, 2, 1, 0	switches
OPCODE1	SW6, 5, 4	3-digit opcodes
OPCODE2	SW6, 5, 4, 3, 2, 1	6-digit opcodes
DATA	SW3, 2, 1, 0	data you use as input to the register
ALU	ALU3, 2, 1, 0	7-segment display
STACK	STACK3, 2, 1, 0	digital display

Adding Signal Groups to Your Display List

In this section, you use the Display Signals dialog box to select the signals to view and debug. To add the groups you just defined to your display list, follow these steps:

1. Select **Settings →Display Signals** from the Debug menu to display the Display Signals dialog box.
2. Use the Display Signals dialog box to include the SWITCHES, ALU, OPCODE1, DATA, and STACK groups to your list of Displayed Signals, as shown in Figure 10-12.
3. Click on the Groups radio button in the Groups group box to display the available signal groups that you just defined.
4. Click on the double right arrow (>>) to move the Available Groups to the Displayed Signals list, as shown in Figure 10-12.

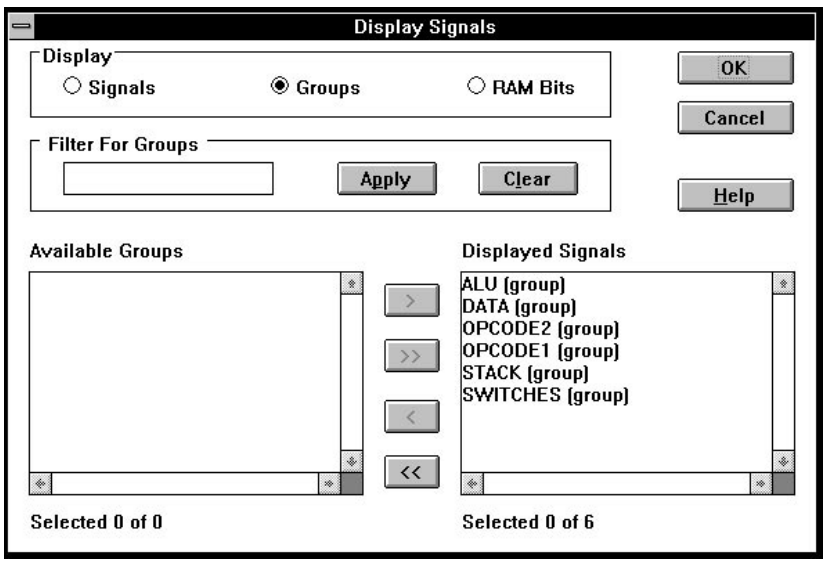


Figure 10-12 Display Signals Dialog Box

5. Remove the OPCODE2 signal group from the list of Displayed Signals by selecting it and clicking on the left arrow (<) to move it to the list of Available Groups. This signal group will be used later in the tutorial.

Adding the EIN Signal to the Display List

You still need to add the EIN signal to your display list. The EIN signal registers the state of the EXC_P switch.

1. Click on the Signals radio button in the Display group box of the Display Signals dialog box to view the list of available signals.
2. Select the EIN signal from the Available Signals list.
3. Click on the right arrow (>) to move the signal to the Displayed Signals column.
4. Click on the **OK** button after adding all the signals.

Reading the Device States

To read the states of the device, follow these steps.

1. Set the opcode switches (SW3-1 through SW3-8) High on your device.
2. Select the **New Waveform** button from the Control panel to open a new waveform display window.
3. Select **Read** from the Control panel to read back a snapshot from the demonstration board and display the values for the selected signals on the waveform display.

   When you read the device, the switches inputs are fed to the SW7 module flip-flops. The oscillator clock — OSC_4K— moves the input values to the flip-flops of the SW7 module. The values you read back reflect the current opcode and data you defined.

   The clock does not affect the values of the SW7 module flip-flops unless you change their inputs by modifying the switches settings. The inputs appear on the 7-segment display (ALU group) only once the EIN output is High. EIN is High when you toggle EXC High and Low again (EXC corresponds to switch 1 on your demonstration board). By setting the opcode correctly, you can push the value currently stored in the register to the stack.

Changing the Signals Groups Radix

To ease the legibility of the waveform signal values, you can modify the radices of the signal groups.

1. Select **Change Radix** from the View menu to change the radices of the DATA and ALU groups to HEX.
2. Click on the radio button for the Hex radix.

3. Highlight the DATA and ALU groups on the Change Radix dialog box.

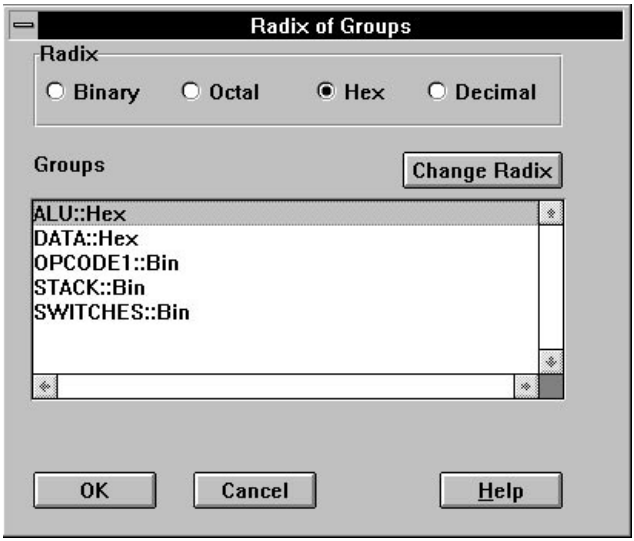


Figure 10-13 Changing the Radix of the ALU and DATA Groups

4. Click on the **`Change Radix`** button.

   The radix that is displayed for DATA and ALU is HEX.

   The radices of the other groups are still binary.

Debugging Example

The waveform in this section represents the states of your switches mapped to the EIN signal and to the signal groups OPCODE1, DATA, ALU, STACK, and SWITCHES.

- EIN is the signal output by the EXC signal.
- OPCODE1 is the group associated with the set of 3-digit opcodes listed in Table 10-8.
- DATA is the group associated with the last four switches on your demonstration board bank of input switches.

- ALU is the group associated with the demonstration board 7-segment display.
- STACK is the group associated with the digital display, which is the right-most row of LED bars on your demonstration board.
- SWITCHES is the group associated with the bank of eight input switches on your demonstration board.

Refer to Figure 10-14 to locate the demonstration board components.

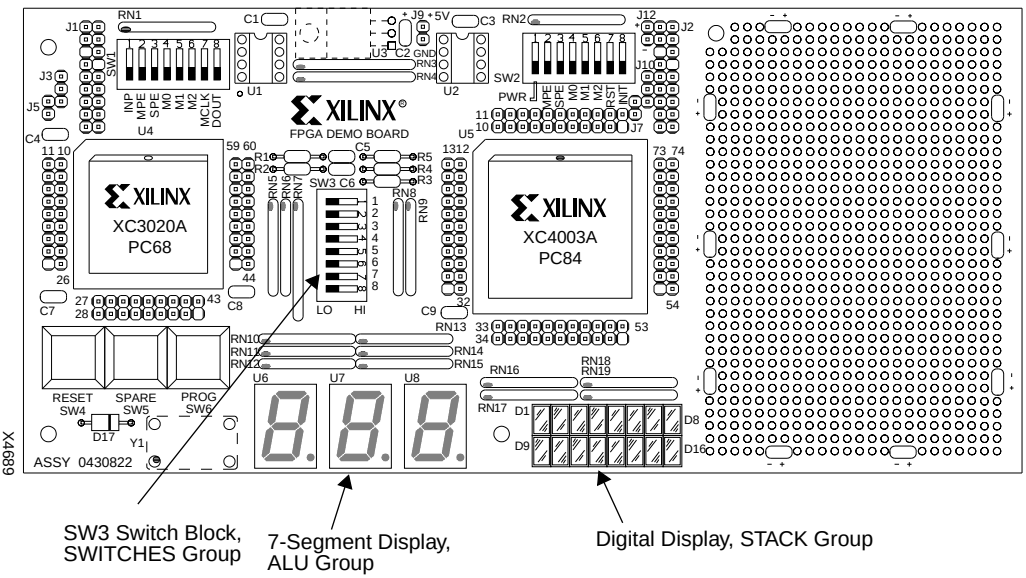


Figure 10-14 FPGA Demonstration Board Components

Capturing the Add Operation

In this section, you add the value 2 to your 7-segment display (ALU group) by setting the switches for OPCODE1 and DATA.

1. Set the OPCODE and DATA switches to the Low position.
2. Select **Read** from the Control panel to read the state of the device.

   The state corresponds to snapshot #1 in Figure 10-15.

Note: Your results may vary from those in Figure 10-15 if your target FPGA is not reset prior to starting this example.

3. Set your input switches 2, 3, and 4 to **000** (ADD opcode) and set switches 7 and 8 to **10** (binary for 2).
4. Select **Read** from the Control panel to read the state of the device.

   The state corresponds to snapshot #2 in Figure 10-15. (State numbers are listed beneath the horizontal axis on the waveform window.)

   The data value 2 appears in the DATA signal group in hexadecimal notation and in the SWITCHES group in binary notation (10).

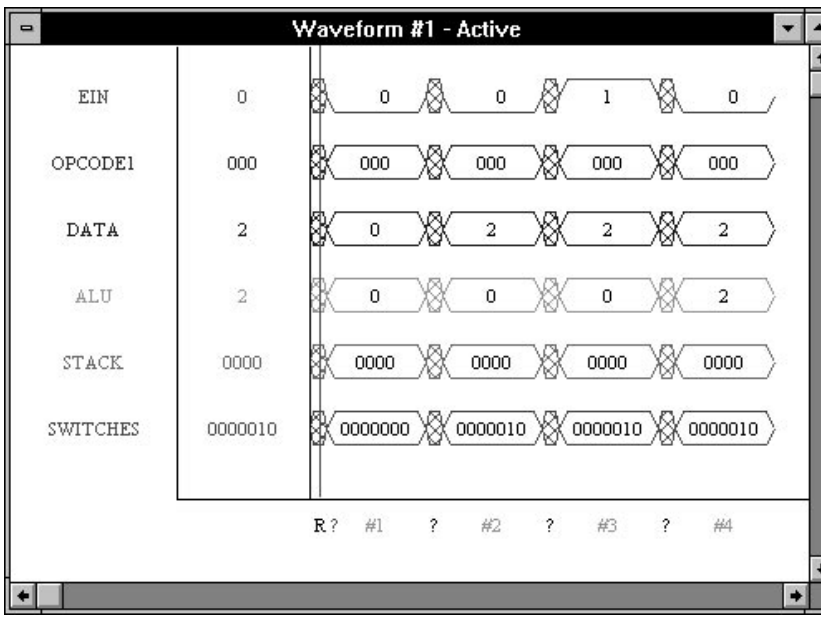


Figure 10-15 Capturing the Device States for the Add Operation

5. Toggle input switch 1 to the High position.
6. Select **Read** from the Control panel to read the state of the device.

   The state corresponds to snapshot #3 in Figure 10-15. The EIN signal is High.
7. Toggle your input switch 1 to the Low position.
8. Select **Read** from the Control panel to read the state of the device.

9. The state corresponds to snapshot #4 in Figure 10-15.

   The 7-segment display (ALU group) displays the data: 2.

Adding and Removing Signal Groups

In this section, you add the OPCODE2 group to your display list and you remove the DATA group, which has no pertinent value in the next part of the tutorial. To bring up the list of defined groups, follow these steps:

1. Click on **`Display Signals`** from the Debug-Settings menu.

   The Display Signals dialog box is displayed

2. Click on the **`Group`** radio button.

   The list of groups already included in your display list appears in the right column of the dialog box., as shown in Figure 10-16.

   The groups that are defined but are not included in the display list are in the left column.

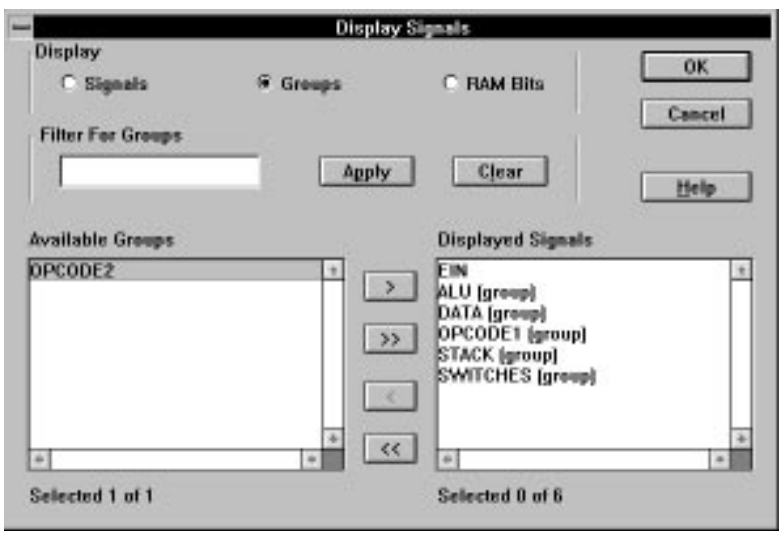


Figure 10-16 Including the OPCODE2 Group in the Display List

Adding a Signal Group to the Display List

To include the OPCODE2 group in your display list, follow these steps:

1. Select the OPCODE2 group in the left column.
2. Click on the single arrow button to move the group to the Displayed Signals column.

   The group will be displayed in the waveform when you quit the dialog box.

Removing a Signal Group from the Display List

To remove the DATA group from the display list, follow these steps:

1. Click on the group DATA in the Displayed Signals list box.
2. Click on the single arrow button to move the signal to the Available Groups list box.
3. Click on the Display Signals dialog box **OK** button to quit the current dialog box and save the changes.

Updating the Waveform

To view the changes you made to the list of displayed signals. read the states of the device.

4. Select **Read** from the Control panel to read the state of the device.
5. The waveform is updated as shown in Figure 10-17. The DATA group has been removed from the waveform and that OPCODE2 has been added to the waveform. OPCODE2 appears in cross-hatched boxes.

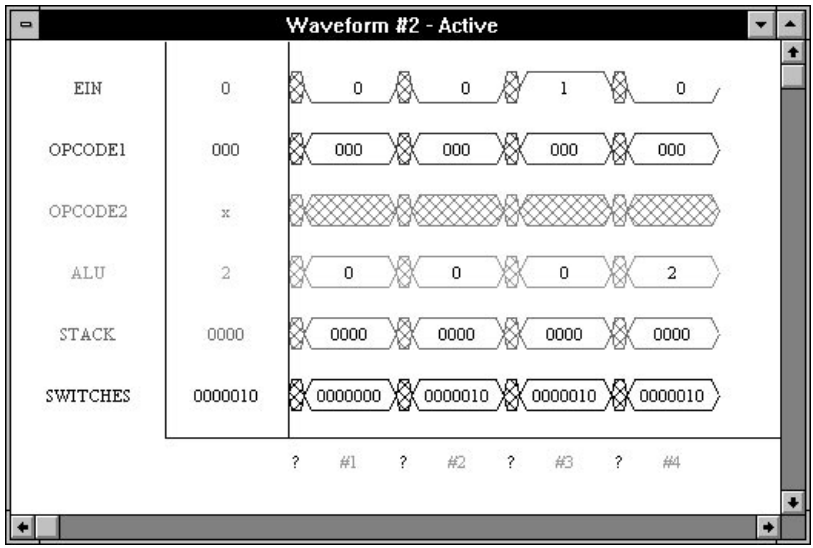


Figure 10-17 Adding and Removing Groups from the Waveform

Capturing the Push Operation

To push the values displayed on the 7-segment display to the stack, follow these steps:

1. Set your input switches 2, 3, 4, 5, 6, and 7 to **`111101`** — PUSH opcode.
2. Select **`Read`** from the Control panel to read the state of the device.

   The state corresponds to snapshot #5 in Figure 10-18. The SWITCHES and OPCODE2 groups are updated to reflect the new switch settings.
3. Toggle your input switch 1 to the High position.
4. Select **`Read`** from the Control panel to read the state of the device.

   The state corresponds to snapshot #6 in Figure 10-18. The EIN signal is High.

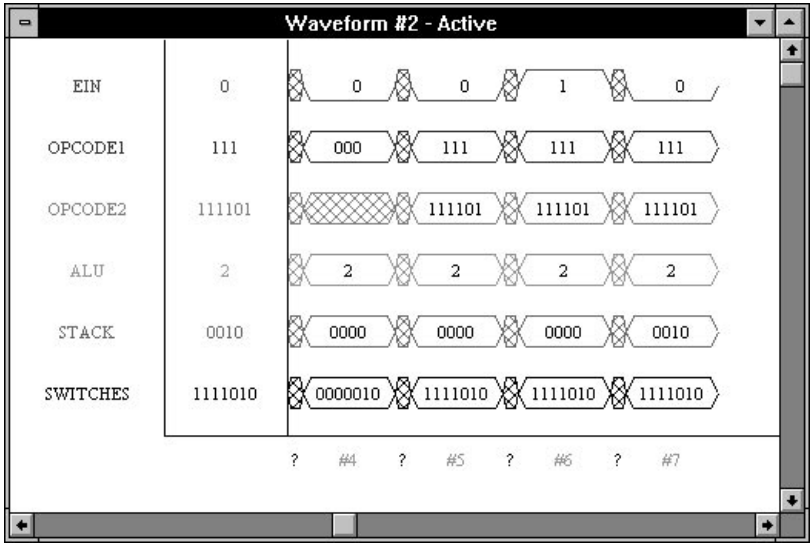


Figure 10-18 Capturing the Device States for the Push Operation

5. Toggle your input switch 1 to the Low position.
6. Select **Read** from the Control panel to read the state of the device.

   The state corresponds to snapshot #7 in Figure 10-18. The STACK group reflects the state of the 7-segment display (ALU group) in binary notation.

Appendix A

Diagnostics and Board Compatibility

XChecker includes a test fixture that you use to test the XChecker cable and a 3 V adapter that you can use to support 3 V target boards. This appendix explains how to test the XChecker cable and how to install and test the 3 V adapter.

Testing the XChecker Cable

The XChecker cable contains special hardware that enables it to perform readback of FPGA devices. To verify that the cable is in good working order, run the Self-Check option from the Cable menu.

Completing a Cable Self-Check

1. Insert the test fixture, provided with your XChecker cable, onto the XChecker cable pins.

   The test fixture is a small printed circuit card with a keyed header connector that fits onto the XChecker cable pins.

2. XChecker draws power from your target system, not from the host system. To this effect, the test fixture has two connectors: a red V_{CC} power (+5 V) connector and a black Ground connector. Plug these connectors to a 5 V DC power supply.

3. Start the Hardware Debugger software or, if the Hardware Debugger was running when you installed the test fixture, click on **`Reset`** from the Cable menu to establish communication with the cable.

4. Select **`Self-Check`** from the Cable menu to perform a diagnostic of the XChecker cable.

   The command invokes a dialog box with the message "`Place Test Fixture.`"

5. Select **OK** to start the diagnostic.

   The output is displayed in a status window.

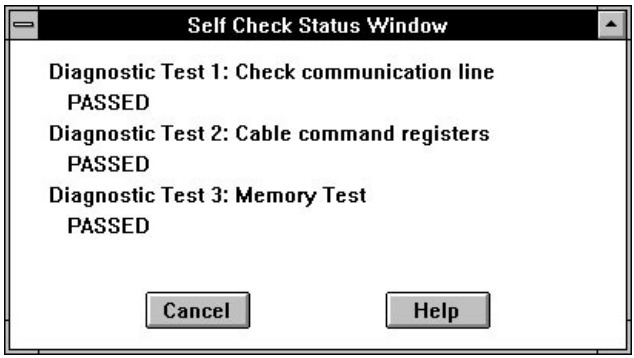


Figure A-1 Self-Check Status Output

Using the 3 V Adapter

The 3 V adapter supports V_{CC} supply voltages from your target system that range from 2.9 V to 5.25 V. An internal voltage 'step-up' circuit generates the 5 V voltage supply needed by the XChecker cable.

Aside from the voltage conversion, the 3 V adapter is completely transparent to the XChecker hardware or the target system. Therefore, you do not need to remove the 3 V adapter when moving the XChecker cable between 3 V and 5 V systems.

The 3 V adapter includes two connectors, J1 and J2.

Connecting the 3 V Adapter

If you are using a 3 V target board instead of a 5 V target board, you must connect the 3 V adapter to the XChecker cable.

Warning: Use normal electro-static discharge (ESD) precautions when connecting the adapter to your XChecker cable, as the adapter is static sensitive and can be damaged by ESD energy.

1. Ensure that you are adequately grounded before connecting or using the 3 V adapter.

2. Referring to the next figure, position the adapter on top of the XChecker assembly aligning the adapter 18-pin female connector (J2) with the XChecker 18-pin connector socket.

   Ensure the Xilinx logo on the XChecker case and the 3 V adapter silkscreen are oriented the same way.

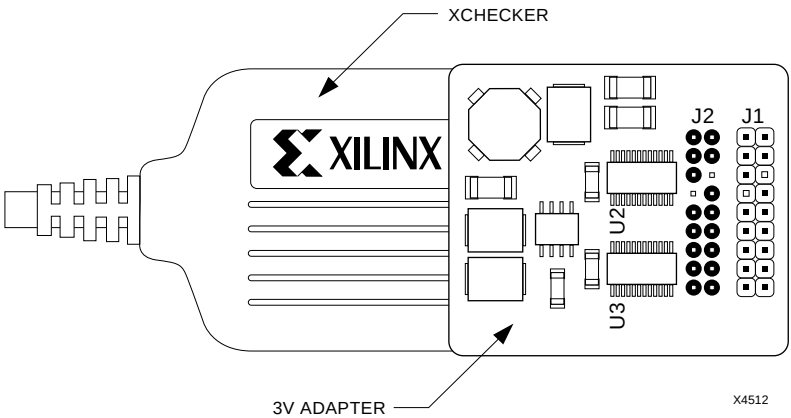


Figure A-2 XChecker 3-Volt Adapter

3. Holding the adapter board by the edges, press it down until it makes a solid connection to the XChecker cable.

Verifying 3 V Adapter Operation

To test the operation of the 3 V adapter, use the test fixture that is shipped with the XChecker cable.

1. Plug the adapter onto the XChecker case, as outlined in the previous section.
2. Plug the test fixture onto the male connector (J1) of the adapter.

   The test fixture is a small printed circuit card with a keyed header connector that fits onto the XChecker cable pins.

3. XChecker draws power from your target system, not from the host system. To this effect, the test fixture has two connectors: a red V_{CC} power (+3 V) connector and a black Ground connector. Plug these connectors to a 3 V DC power supply.

4. Start the Hardware Debugger software or, if the Hardware Debugger was running when you installed the test fixture, click on **Reset** from the Cable menu to establish communication with the cable.
5. Select **Self-Check** from the Cable menu to perform a diagnostic of the XChecker cable.
6. Select **OK** to start the diagnostic.

   This command runs the diagnostics test. The test should produce results that match those obtained by running the same diagnostics test without the 3 V adapter connected to XChecker. See Figure A-1.

Note: The readback clock speed of the XC3000L devices has been slowed because of lower V_{CC} supply voltage. If the supply voltage of the target system is lower than 3 V, you might see error messages like the following when reading back or verifying a configured XC3000L device:

```
Design design_name has 128 probeable signals.
Readback 1847 bytes of configuration.
Verifying datafile design_name...MISMATCHED
Total of 405 bits mismatched.
```

Should this occur, correct the system voltage and redownload the design.

Appendix B

Console Commands

Console commands refer to commands entered from the Console window. You can use console commands as an alternative to menu commands. By default, console commands do not display the dialog and message boxes that menu commands invoke. To get command feedback, you can use the Force command. Additionally, you can also turn on the Show Command and Its Status check box in the Console window.

To automate design debugging, you can copy commands from the Console window into a macro window to build macros. See the "Customizing the Interface" chapter for information.

Note: You should not use console commands until you are familiar with the graphical user interface commands, as the software expects these commands to be executed in a particular order.

Conventions

The following is a summary of the syntax used for Console commands.

- When an option occurs without parentheses next to the command name in the syntax, it is required. When it appears in square brackets, the variable is optional.
- When two or more options occur between curly brackets, it means they must be entered with the command. If the options are separated by a pipe character, it means you must choose one of the possible parameters. If one term is broken down into a subset of parameters that can be entered separately or together, each of these subparameters occurs between square brackets.

Alphabetical Listing of Commands

The following summarizes the console commands discussed in this chapter.

Baud	Sets the baud rate
Cable	Sets the cable options
Clock	Sets the clock options
Display	Selects the signals to debug
Download	Programs target device with the current design
Exit	Exits the Hardware Debugger
Force	Displays or hides the message boxes
Group	Defines groups of signals for debugging
Open	Loads the configuration data
Port	Selects the communications port
Quit	Exits the Hardware Debugger
Readfpga	Reads the device states using the specified debugging settings (snapshot number, signals, and groups)
Reset	Reinitializes the target device
Run	Executes the macro in the current macro window
Setmode	Selects the debugging mode, synchronous or asynchronous
Trigger	Selects the source of the readback trigger
Verify	Verifies the design was downloaded correctly

Baud

The Baud command sets the cable baud rate.

Syntax

The syntax of the Baud command is the following:

Baud *rate*

Parameters

The Baud command has one parameter:

- *rate* specifies the rate at which the system is run. This parameter can be set to {9600 | 19200 | 38400}

Examples

Following is an example of how to specify the Baud command.

```
Baud 38400
```

Abbreviations

Baud **bau**

Cable

The Cable command is twofold. On the one hand, it enables you to set the cable options. On the other hand it informs the Hardware Debugger of the type of cable you have installed.

Syntax

The syntax of the Cable command is the following:

Cable {*option* | *name*}

Parameters

The variables of the Cable command are further broken down into the following parameters:

- *option* {–reset | –check | –pins}

- –Reset reprograms the XChecker cable.
- –Check tests the cable operation.
- –Pins checks the logic levels of the XChecker pins.

- *name* {xchecker | serial | parallel}
  - Xchecker establishes communication with the XChecker cable.
  - Serial establishes communication with a serial cable.
  - Parallel establishes communication with a parallel cable.

Examples

Following are examples of how to specify the Cable command.

```
Cable –pins
Cable –xchecker
```

Abbreviations

You can abbreviate the Cable command and its options as follows:

Cable	**cab**
–Reset	**–rs**
–Check	**–chk**
–Pins	**–pn**

Clock

The Clock command is a synchronous mode debugging option. The command is twofold. On the one hand it enables you to specify the clock options, namely the clock type and speed. On the other hand, it enables you to specify the number of clocks to apply to the target device during debugging.

Syntax

The syntax of the Clock command is the following:

Clock { [*source*][*speed*] | *apply_clocks* | *control_option*}

Parameters

The variables of the Clock command are further broken down into the following parameters:

- *source* {–internal | –external}
  - –Internal sets the clock as internal
  - –External sets the clock as external
- *speed* –speed {1 | 3 | 5 | 11}
  - –Speed sets the speed of the clock.
- *apply_clocks* –apply {32-bit number}
  - –Apply cycles the device forward the number of clocks you specify
- *control_option* {–stop | –resume}
  - –Stop interrupts the system clock
  - –Resume resumes use of the system clock

Examples

Following are examples of how to specify the Clock command.

```
Clock –internal –speed 3
Clock –apply 4
Clock –stop
```

Abbreviations

You can abbreviate the Clock command and its options as follows:

Clock	**clk**
–Internal	**–int**
–External	**–ext**
–Apply	**–ap**
–Stop	**–st**
–Resume	**–res**

Display

The Display command specifies which nets to debug.

Syntax

The syntax of the Display command is the following:

Display *operation list_of_signals*

Parameters

The variables of the Display command are further broken down into the following parameters:

- *operation* {–del | –add}
  - –Del deletes the specified list of signals and/or group names from the display list
  - –Add adds the specified signals and group names to the display list
- *list_of_signals* { [*list_of_signals*] [*group_names*] | *}
  - *list_of_signals* is the list of signals you wish to probe
  - *group_names* is the list of defined signal groups you wish to probe
  - * is the complete list of signals and groups in your design if you are adding signals. It refers to the signals in the display list if you are removing signals.

Examples

Following are examples of how to specify the Display command.

```
Display –add A B C D E F G ALUOUT SWITCHES
Display –del *
```

Abbreviations

You can abbreviate the Display command as follows:

Display **dply**

Download

The Download command enables you to program a device by downloading the current design to that device.

Syntax

The syntax of the Download command is the following:

Download [**–***verify*]

Parameters

The Download command has one parameter:

–Verify reads back the data that was downloaded to the target device and compares it to the original data.

Examples

Following is an example of how to specify the Download command.

```
Download –verify
```

Abbreviations

You can abbreviate the Download command as follows:

Download	**dn**
–Verify	**–v**

Exit

The Exit command enables you to exit the Hardware Debugger program.

Syntax

The syntax of the Exit command is the following:

Exit

Parameters

The Exit command has no parameters.

Examples

Following is an example of how to specify the Exit command.

```
Exit
```

Abbreviations

You can abbreviate the Exit command as follows:

Exit	**ex**

Force

The Force command enables you to display or hide information dialog boxes while a macro is running.

Syntax

The syntax of the Force command is the following:

Force *setting*

Parameters

The parameters of the Force command are the following:

- *setting* {–on | –off}
  - –On displays the information dialog boxes
  - –Off hides the information dialog boxes

Examples

Following is an example of how to specify the Force command.

```
Force –on
```

Abbreviations

You can abbreviate the Force command and its options as follows:

Force	**for**
–On	**–o**
–Off	**–f**

Group

The Group command is twofold. On the one hand it enables you to specify signal groups to debug as entities. On the other hand, it deletes existing groups of signals.

Syntax

The syntax of the Group command is the following:

Group {*del_group* | *new_group*}

Parameters

The variables of the Group command can be broken down into the following parameters:

- where *del_group* is –del *groupname list_of_signals*
  - –Del deletes the specified signals from the specified group name
  - *groupname* is the name of the group you are defining
  - *list_of_signals* is the list of signals you are adding or removing from the group
- where *new_group* is *groupname list_of_signals* and adds the specified signals and group names to the display list
  - *groupname* is the name of the group you are defining
  - *list_of_signals* is the list of signals you are adding or removing from the group

Examples

Following is an example of how to specify the Group command.

```
Group aluout ALU0 ALU1 ALU2 ALU3 ALU4
Group –del aluout ALU2
```

Abbreviations

You can abbreviate the Group command as follows:

Group **gp**

Open

The Open command opens a file for downloading.

Syntax

The syntax of the Open command is the following:

Open *file*

Parameters

The variables of the Open command can be broken down into the following parameters:

- *file* {*filepath**file*}
  - *filepath* is the drive and directory in which your design is located
  - *filename* is the name of the design you want to open

Examples

Following is an example of how to specify the Open command.

```
Open C:\TMP\4KACALC\CALC4K.BIT
```

Abbreviations

You can abbreviate the Open command as follows:

Open	**op**

Port

The Port command enables you to select the communications port you need for your cable.

Syntax

The syntax of the Port command is the following:

Port *portname*

Parameters

The parameters of the Port command are the following:

- where *portname* is

  {com1 | com2 | com3 | com4}
  (for XChecker or Serial cables)

  or

  {lpt1 | lpt2 | lpt3 | lpt4}
  (for Parallel cable only)

Examples

Following are examples of how to specify the Port command.

```
Port com1
Port lpt2
```

Abbreviations

You can abbreviate the Port command as follows:

Port	**po**

Quit

The Quit command enables you to quit the Hardware Debugger.

Syntax

The syntax of the Quit command is the following:

```
Quit
```

Parameters

There are no parameters for the Quit command.

Examples

Following is an example of how to specify the Quit command.

```
Quit
```

Abbreviations

You can abbreviate the Quit command as follows:

Quit	**qt**

Readfpga

The Readfpga command enables you to debug the states of a device.

Syntax

The syntax of the Readfpga command is the following:

Readfpga *snapshots*

Parameters

The parameters of the Readfpga command are *snapshots* {1–65534}

Examples

Following is an example of how to specify the Readfpga command.

```
Readfpga 4
```

Abbreviations

You can abbreviate the Readfpga command as follows:

Readfpga	**rea**

Reset

The Reset command enables you to reset the FPGA.

Syntax

The syntax of the Reset command is the following:

Reset

Parameters

There are no parameters for the Reset command.

Examples

Following is an example of how to specify the Reset command.

```
Reset
```

Abbreviations

You can abbreviate the Reset command as follows:

Reset **res**

Run

Executes the macro defined in the current macro window.

Syntax

The syntax of the Run command is the following:

Run

Parameters

There are no parameters for the Run command.

Examples

Following is an example of how to specify the Run command.

```
Run
```

Abbreviations

You can abbreviate the Run command as follows:

Run **rn**

Setmode

The Setmode command enables you to set the debugging mode.

Syntax

The syntax of the Setmode command is the following:

Setmode *debugmode*

Parameters

The parameters of the Setmode command are the following:

- *debugmode {–sync | –async}*
  - –Sync turns the synchronous mode debugging on
  - –Async turns the asynchronous mode debugging on

Examples

Following is an example of how to specify the Setmode command.

```
Setmode -sync
```

Abbreviations

You can abbreviate the Setmode command and its options as follows:

setmode	**md**
–Synch	**–s**
–Async	**–as**

Trigger

The Trigger command specifies the source of the readback trigger.

Syntax

The syntax of the Trigger command is the following:

Trigger {*timing* | *clocksettings* | *option*}

Parameters

The parameters of the Trigger command are the following:

- *timing* {–immediately | –external | –enterkey} sets the trigger type
- *clocksettings* –clock {before_first_snapshot between_snapshots} specifies the clock patterns to apply to the target device.

- *option* {–timeout *num* | –reset}
  - {–timeout *num}* specifies the cutoff time for a trigger to be received
  - –Reset reinitializes the target device

Examples

Following is an example of how to specify the Trigger command.

```
Trigger –immediately –clock 2 2 –timeout 10
–reset
```

Abbreviations

You can abbreviate the Trigger command as follows:

Trigger	**tr**

Verify

The Verify command enables you to verify that the design was downloaded correctly.

Syntax

The syntax of the Verify command is the following:

```
Verify
```

Parameters

The Verify commands has no parameters.

Examples

Following is an example of how to specify the Verify command.

```
Verify
```

Abbreviations

You can abbreviate the Verify command as follows:

Verify	**ver**

Index

D

Trademark Information