XILINX®

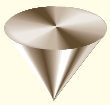
ONLINE

EZTAG

USER GUIDE

TABLE OF CONTENTS

INDEX



GO TO OTHER BOOKS

XSTEP™

0401405

EZTag User Guide

Preface

About This Manual

This manual describes Xilinx's EZTag software, a tool used for In-system progamming.

Before using this manual, you should be familiar with the operations that are common to all Xilinx's software tools: how to bring up the system, select a tool for use, specify operations, and manage design data. These topics are covered in the *Development System Reference Guide*.

Manual Contents

This manual covers the following topics.

- Chapter 1, "Introduction," provides an introduction to Xilinx Boundary-scan and JTAG capabilities.
- Chapter 2, "EZTag Download Cable Options," provides information for connecting and using the XChecker Serial Cable or the Parallel Download Cable to download and readback information in-system.
- Chapter 3, "In-System Tutorial for PCs," documents the basics of using the interface to download programming to XC9500 family devices in-system.
- Chapter 4, "EZTag with Workstations," documents the basics of using EZTag with from a workstation environment.
- Appendix A, "Error Messages," provides a list of error messages that EZTag may report. For most error messages a workaround is suggested.

Conventions

In this manual the following conventions are used for syntax clarification and command line entries.

- `Courier font` indicates messages, prompts, and program files that the system displays, as shown in the following example.

  `speed grade: -100`

- **`Courier bold`** indicates literal commands that you must enter in a syntax statement.

  `rpt_del_net=`

- *Italic font* indicates variables in a syntax statement. See also, other conventions used on the following page.

  `xdelay` *design*

- Square brackets "[ ]" indicate an optional entry or parameter. However, in bus specifications, such as bus [7:0], they are required.

  `xdelay [`*option*`]` *design*

- Braces "{ }" enclose a list of items from which you choose one or more.

  `xnfprep` *designname* **`ignore_rlocs={true|false}`**

- A vertical bar " | " separates items in a list of choices.

  `symbol` *editor* `[bus|pins]`

Other conventions used in this manual include the following.

- *Italic font* indicates references to manuals, as shown in the following example.

  See the *Development System Reference Guide* for more information.

- *Italic font* indicates emphasis in body text.

  If a wire is drawn so that it overlaps the pin of a symbol, the two nets *are not* connected.

- A vertical ellipsis indicates repetitive material that has been omitted.

  ```
  IOB #1: Name = QOUT'
  IOB #2: Name = CLKIN'
  .
  .
  .
  ```

- A horizontal ellipsis "..." indicates that the preceding can be repeated one or more times.

  `allow block` *blockname loc1 loc2* `...` *locn* `;`

Contents

Chapter 1

Introduction

This chapter introduces you to the basic concepts of Xilinx JTAG capabilities and the XC9500 series products.

Boundary Scan

What is IEEE 1149.1

Design complexity, difficulty of loaded board testing, and the limited pin access of surface mount technology led industry leaders to seek accord on a standard to support the solution of these problems.

Boundary Scan, formally known as IEEE Standard 1149.1, is primarily a testing standard created to alleviate the growing cost of designing and producing digital systems. The primary benefit of the standard is the ability to transform extremely difficult printed circuit board testing problems (that could only be attacked with ad-hoc testing methods) into well-structured problems that software can handle easily and swiftly.

The standard defines a hardware architecture and the mechanisms for its use to solve the aforementioned problems.

What can it be used for

Although primarily a testing standard for on-chip circuitry, the proliferation of the standard has opened to the door to a wide variety of applications. The standard itself defines instructions that can be used to perform functional and interconnect tests as well as built-in self test procedures.

Vendor-specific extensions to the standard have been developed to allow execution of maintenance and diagnostic applications as well as programming algorithms for reconfigurable parts. It is the latter

that have been implemented (in addition to all the mandatory operations of the standard and some optional ones) in the FastFLASH family.

How does it work

The top level schematic of the test logic defined by IEEE Std 1149.1 includes three key blocks:

The TAP Controller

This responds to the control sequences supplied through the test access port (TAP) and generates the clock and control signals required for correct operation of the other circuit blocks.

The Instruction Register

This shift register-based circuit is serially loaded with the instruction that selects an operation to be performed.

The Data Registers

These are a bank of shift register based circuits. The stimuli required by an operation are serially loaded into the data registers selected by the current instruction. Following execution of the operation, results can be shifted out for examination.

The JTAG Test Access Port (TAP) contains four pins that drive the circuit blocks and control the operations specified. The TAP facilitates the serial loading and unloading of instructions and data. The four pins of the TAP are: TMS, TCK, TDI and TDO. The function of each TAP pin is as follows:

TCK - this pin is the JTAG test clock. It sequences the TAP controller as well as all of the JTAG registers provided in the XC95108.

TMS - this pin is the mode input signal to the TAP Controller. The TAP controller is a 16-state FSM that provides the control logic for JTAG. The state of TMS at the rising edge of TCK determines the sequence of states for the TAP controller. TMS has an internal pull-up resistor on it to provide a logic 1 to the system if the pin is not driven.

TDI -this pin is the serial data input to all JTAG instruction and data registers. The state of the TAP controller as well as the particular instruction held in the instruction register determines which register

is fed by TDI for a specific operation. TDI has an internal pull-up resistor on it to provide a logic 1 to the system if the pin is not driven. TDI is sampled into the JTAG registers on the rising edge of TCK.

TDO - this pin is the serial data output for all JTAG instruction and data registers. The state of the TAP controller as well as the particular instruction held in the instruction register determines which register feeds TDO for a specific operation. Only one register (instruction or data) is allowed to be the active connection between TDI and TDO for any given operation. TDO changes state on the falling edge of TCK and is only active during the shifting of data through the device. This pin is three-stated at all other times.

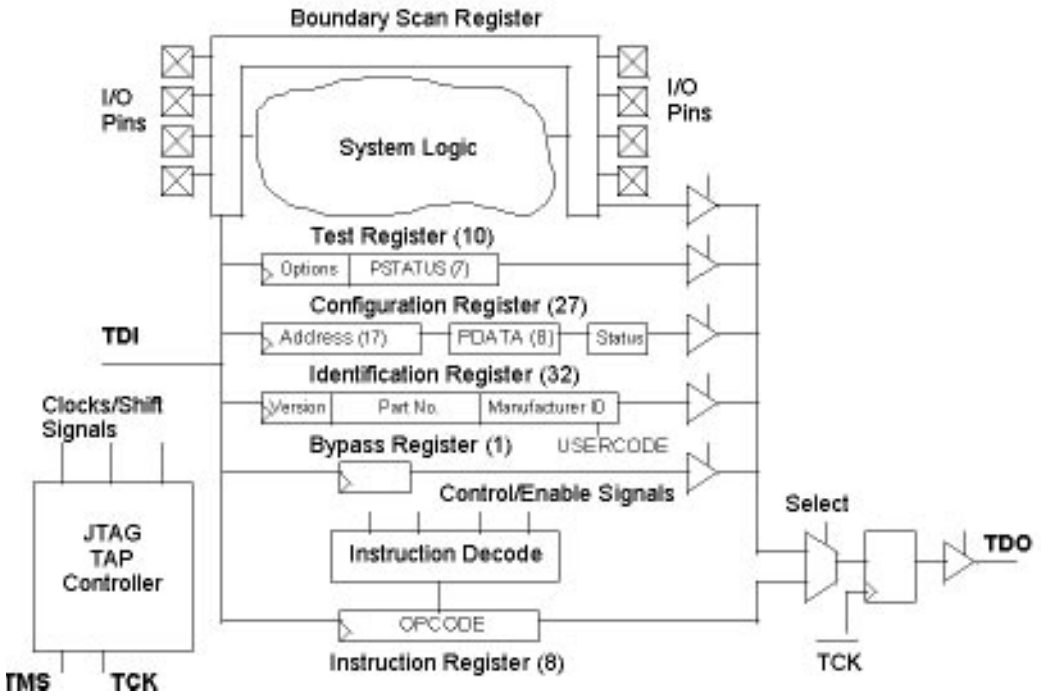


Figure 1-1 JTAG Architecture

JTAG TAP Controller

The JTAG TAP Controller is a 16-state finite state machine, that controls the scanning of data into the various registers of the JTAG architecture. A state diagram of the TAP controller is shown in Figure

1-1. The state of the TMS pin at the rising edge of TCK is responsible for determining the sequence of state transitions. There are two state transition paths for scanning the signal at TDI into the device, one for shifting in an instruction to the instruction register and one for shifting data into the active data register as determined by the current instruction.

JTAG TAP Controller States

Test-Logic-Reset. This state is entered on power-up of the device whenever at least five clocks of TCK occur with TMS held high. Entry into this state resets all JTAG logic to a state such that it will not interfere with the normal component logic, and causes the IDCODE instruction to be forced into the instruction register.

Run-Test-Idle. This state allows certain operations to occur depending on the current instruction. For the XC9500 family, this state causes generation of the program, verify and erase pulses when the associated in-system programming (ISP) instruction is active.

Select-DR-Scan. This is a temporary state entered prior to performing a scan operation on a data register or in passing to the Select-IR-Scan state.

Select-IR-Scan. This is a temporary state entered prior to performing a scan operation on the instruction register or in returning to the Test-Logic-Reset state.

Capture-DR. This state allows data to be loaded from parallel inputs into the data register selected by the current instruction on the rising edge of TCK. If the selected data register does not have parallel inputs, the register retains its state.

Shift-DR. This state shifts the data, in the currently selected register, towards TDO by one stage on each rising edge of TCK after entering this state.

Exit1-DR. This is a temporary state that allows the option of passing on to the Pause-DR state or transitioning directly to the Update-DR state.

Pause-DR. This is a wait state that allows shifting of data to be temporarily halted.

Exit2-DR. This is a temporary state that allows the option of passing on to the Update-DR state or returning to the Shift-DR state to continue shifting in data.

Update-DR. This state causes the data contained in the currently selected data register to be loaded into a latched parallel output (for registers that have such a latch) on the falling edge of TCK after entering this state. The parallel latch prevents changes at the parallel output of these registers from occurring during the shifting process.

Capture-IR. This state allows data to be loaded from parallel inputs into the instruction register on the rising edge of TCK. The least two significant bits of the parallel inputs must have the value 01 as defined by IEEE Std. 1149.1, and the remaining 6 bits are either hard-coded or used for monitoring of the security and data protect bits.

Shift-IR. This state shifts the values in the instruction register towards TDO by one stage on each rising edge of TCK after entering this state.

Exit1-IR. This is a temporary state that allows the option of passing on to the Pause-IR state or transitioning directly to the Update-IR state.

Pause-IR. This is a wait state that allows shifting of the instruction to be temporarily halted.

Exit2-IR. This is a temporary state that allows the option of passing on to the Update-IR state or returning to the Shift-IR state to continue shifting in data.

Update-IR. This state causes the values contained in the instruction register to be loaded into a latched parallel output on the falling edge of TCK after entering this state. The parallel latch prevents changes at the parallel output of the instruction register from occurring during the shifting process.

JTAG Instructions Supported in FastFLASH Parts

Mandatory Boundary Scan Instructions

BYPASS. The BYPASS instruction allows rapid movement of data to and from other components on a board that are required to perform test operations.

SAMPLE/PRELOAD. The SAMPLE / PRELOAD instruction allows a snapshot of the normal operation of a components to be taken and examined. It also allows data values to be loaded onto the latched parallel outputs of the boundary scan shift register prior to the selection of other boundary-scan test instructions.

EXTEST. The EXTEST instruction allows testing of off-chip circuitry and board level interconnections.

Optional Boundary Scan Instructions

INTEST. The INTEST instruction allows testing of the on-chip system logic while the components are already on the board.

HIGHZ. The HIGHZ instruction forces all drivers into high impedance states.

IDCODE. The IDCODE instruction allows blind interrogation of the components assembled onto a printed circuit board to determine what components exist in a product.

USERCODE. The USERCODE instruction allows a user-programmable identification code to be shifted out for examination. This allows the programmed function of the component to be determined.

FastFLASH Reconfiguration Instructions

ISPEN. The ISPEN instruction activates the FastFLASH part for in-system programming.

FPGM. The FPGM instruction is used to program the fuse locations at a specified address.

FERASE. The FERASE instruction is used to perform an erase of a block of fuse locations.

FVFY. The FVFY instruction is used to read the programming of the fuse locations at a specified address.

ISPLD. The ISPLD instruction loads the programmed values into the device memory. It then activates the device to operate according to the programmed values.

Device Operations

Programming information is extracted from the JEDEC file generated by the fitter software. The JEDEC file name defaults to **<*design*>.jed**. If a full pathname is not provided the file of this name is searched for along the XACT path.

Device operation options available to users are:

Program. Download the contents of the JEDEC file to the device programming registers.

Program & Verify. Download contents of the JEDEC file to the device programming registers. Configure the device and read back the contents of device programming registers and compare them with the JEDEC file. Report any differences to the user.

Program, Verify & Test. Same as **Program & Verify** (above) with the addition of a **Functional Test** (see below).

Verify. Read back the contents of the device programming registers and compare them with the JEDEC file.

Erase. Clear device configuration information.

Functional Test. Apply user-specified functional vectors from the JEDEC file to the device, comparing results obtained against expected values. Report any differences to the user.

Read Manufacturer's ID. Read the contents of the JTAG IDCODE register. Display contents for the user.

Read User Signature. This value will have been set by the user at programming time. It is valid only after programming. This function reads the contents of the JTAG USERCODE register and displays the result for the user.

Bypass. Ignore this device when addressing devices in the JTAG boundary scan chain.

Readback. Reads back the contents of device programming registers and creates a new JEDEC file with the results.

Checksum. Reads back the contents of device programming registers and calculates a checksum for comparison against the expected value.

Data Security

All programming operations can optionally select to enable “Data Protection” or Data Security” or both.

When enabled, **`Data Security`** disables reading the programmed contents of a device (the device’s id and signature remain readable).

`Data Protection` allows only the reading of the programmed data. The device contents cannot be altered.

When both **`Data Security`** and **`Data Protection`** are enabled, the device can be neither read nor re-programmed.

Feedback

When using the PC-based graphical user interface, alert boxes at the end of operation execution provide immediate feedback as the success or failure of the specified operation. Detailed information regarding failure is located in the system log file, and is provided for both the PC and workstation based tool.

Disconnecting

Turn off the target system power before disconnecting the FastFLASH BSCAN Download cable or the XChecker cable.

Modifying a Programmed Design File

The flow does not change for a modified programmed design file. There are no shortcuts even with a minor change.

BSDL Summary

The Boundary Scan Description Language (BSDL) uses a subset of VHDL to describe the boundary scan features of a component. One BSDL file is required for each kind of boundary-scan device in the system.

The BSDL files for FastFLASH devices are provided as part of the product release. The user is responsible for providing BSDL files for any non-FastFLASH parts used in the boundary-scan chain.

The system looks for BSDL files along the XACT path and in the current working directory. The name of the BSDL file is assumed to be **<*device name*>.bsd**.

JEDEC 3C Summary

This is the fuse map and functional verification vector file. This is an ASCII file containing the configuration information and, optionally, the vectors that can be used to verify the functional behavior of the configured part. One JEDEC file is generated for each FastFLASH device in the system.

The system looks for the JEDEC files along the XACT path and in the current working directory. The name of the JEDEC file is assumed to be **<*design name*>.jed**, but can be specified exactly by the user.

Chapter 2

EZTag Download Cable Options

This chapter gives specific information about using EZTag with the XChecker Cable and the Parallel Download Cable. You can use EZTag to download, read back, verify design configuration data for any device, and to probe internal logic states of an EPLD design.

Two cables are available from Xilinx. The first is a serial RS-232 known as the XChecker that you can connect to a serial port. The second is the Parallel Download Cable that you can connect to a printer port. This chapter documents both cables.

Note: If you have a Parallel Download Cable proceed to page 2-8.

XChecker Hardware (Serial)

The XChecker hardware consists of a cable assembly with internal logic, a test fixture, and a set of headers to connect the cable to your target system. If you have a serial system you may need a DB-9/DB-25 adapter to connect to the host computer.

Using the XChecker hardware requires either a standard DB-9 or DB-25 RS-232 serial port, or a parallel port. If you have a different serial port connection, you need to provide an appropriate adapter. Figure

2-1 shows the XChecker cable hardware and accessories.

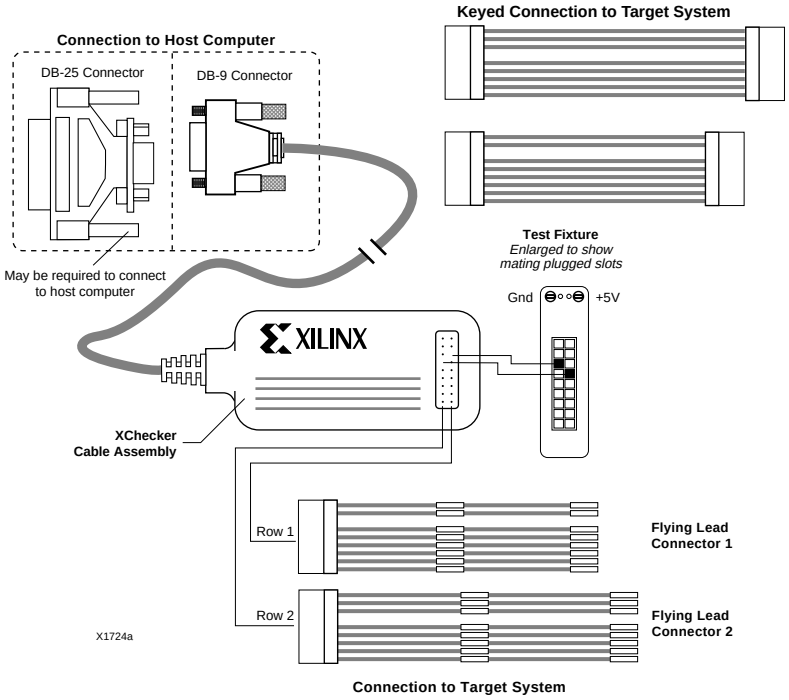


Figure 2-1 XChecker Hardware and Accessories

Figure 2-2 show top and bottom views of the XChecker cable.

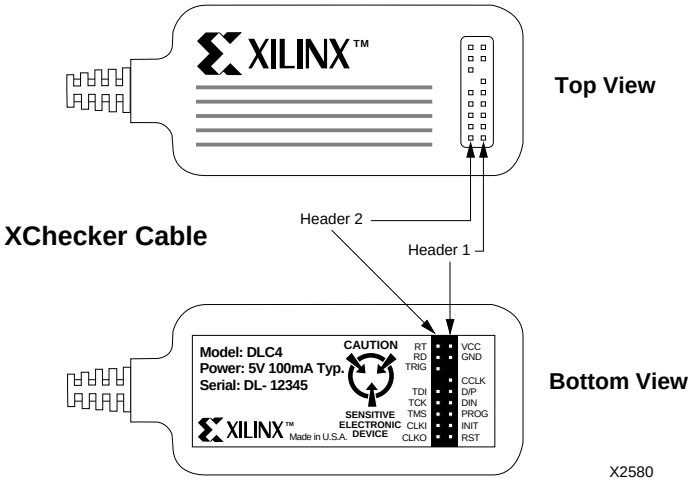


Figure 2-2 XChecker Cable

The cable assembly houses internal circuitry consisting of a Xilinx FPGA and a static RAM. The internal Xilinx FPGA functions as an interface between the XChecker software and the target system. The static RAM stores data for readback.

You can use the XChecker cable with a single EPLD or several connected in a boundary-scan chain to download and readback configuration data.

The XChecker cable transmits configuration data to all target EPLDs at 921 kHz.

Communication between the host system and the XChecker cable is dependent on host system capability. Table 2-1 lists the valid baud rates for serial cables using the supported platforms. Parallel cables will support the transfer rate your system uses.

When the XChecker cable is used to drive a JTAG TAP, a special configuration file “xckjtag.sys” is downloaded automatically to the FPGA in the cable assembly. This configuration file converts the FPGA into a special-purpose JTAG TAP processing unit. Special instructions sent to the FPGA via the serial port sequence the TAP to exercise the target system JTAG circuitry. TDO data is captured in the XChecker’s static RAM. This can then be uploaded to the host via the

serial port for further processing.

Table 2-1 Valid Baud Rates

	Baud Rate			
Platform	9600	19200	38400	115.2K
IBM PC	X	X	X	X
NEC PC	X			
SUN	X	X	X	
HP 700	X	X	X	

X indicates supported baud rate

Connecting the XChecker Cable

There are two simple steps for connecting the cable:

- Connect the cable to your host system serial port.
- Connect the cable to your target system.

Connecting the XChecker Cable to Your Workstation

On workstations the XChecker cable connects to your system RS-232 serial port. You may need a DB-9/DB-25 adapter, which accommodates most serial ports, so that you can connect the XChecker cable to your host system.

Connecting the XChecker Cable to Your PC

On PCs you can connect a serial XChecker cable connects to your system RS232 serial port (you may need a DB9/DB25 adapter). The EZTag software will automatically identify thy XChecker cable when correctly connected to your PC. If you choose to, you may also select this connection manually. To set up a serial port manually:

Cable → Serial Port → [*Comn*]

Connection to Your Target System

You need appropriate pins on the target system for connecting the target system board to the header connection on the cable. These connectors must be standard 0.025–inch square male pins that have dedicated traces to the target system control pins. You can connect to these pins with a header connector or a flying lead connector.

Note: The XChecker cable draws its power from the target system through V_{CC} and GND. Therefore, power to XChecker, as well as to the target system, must be stable. Do not connect any signals before connecting V_{CC} and ground.

Header Connector

When you layout the printed circuit board for use with JTAG in-system programming and testing, a few adjustments will make the process of connecting and downloading easier.

- Provide header pins on your printed circuit board for VCC, GND, TCK, TDI, TMS and TDO.

The header connector is two standard 9-pin (8 signals, 1 key) header connectors that fit 0.025″ square male pins. The pin order is listed in Table 2-2. These header connectors are keyed to assure proper orientation to the cable assembly.

Flying Lead Connectors

The flying lead connector is two flying lead header connectors with eight standard individual female connectors on one end that fit onto 0.025″ square male pins. Each lead is labeled to identify the proper pin connection.

Cable Connections

Connections between the cable assembly and the target system use 6 leads. For connection to JTAG boundary-scan systems you need only ensure that the VCC, GND, TDI, TCK, TMS and RD (TDO) pins are connected.

Once installed properly, the connectors provide power to the cable, allow download and readback of configuration data, and provide for logic probe of device pins.

Each pin has a 100-Ω series resistor. You must provide an external pull-up resistor (approximately 10-50-kΩ) where indicated.

Table 2-2 describes the pin connections to the target circuit board.

Table 2-2 XChecker Cable Connections and Definitions

Name	Function	Connections
VCC	Power – Supplies V_{CC} (5 V, 100 mA, typically) to the cable.	To target system V_{CC}
GND	Ground – Supplies ground reference to the cable.	To target system ground
RD (TDO)	Read Data – Read back data from the target system is read at this pin.	Connect to system TDO pin.
TDI	Test Data In – this signal is used to transmit serial test instructions and data.	Connect to system TDI pin.
TCK	Test Clock – this clock drives the test logic for all devices on boundary-scan chain.	Connect to system TCK pin.
TMS	Test Mode Select – this signal is decoded by the TAP controller to control test operations.	Connect to system TMS pin.
CLKI	Not used.	Unconnected.
CLKO	Not used.	Unconnected.
CCLK	Not used.	Unconnected
D / P	Not used.	Unconnected
DIN	Not used.	Unconnected
PROG	Not used.	Unconnected.
INIT	Not used.	Unconnected.

Name	Function	Connections
RST	Not used.	Unconnected.
RT	Not used.	Unconnected.
TRIG	Not used.	Unconnected.

Connecting for System Operation

Connect the XChecker cable to the host system and your target system as shown in Figure 2-3.

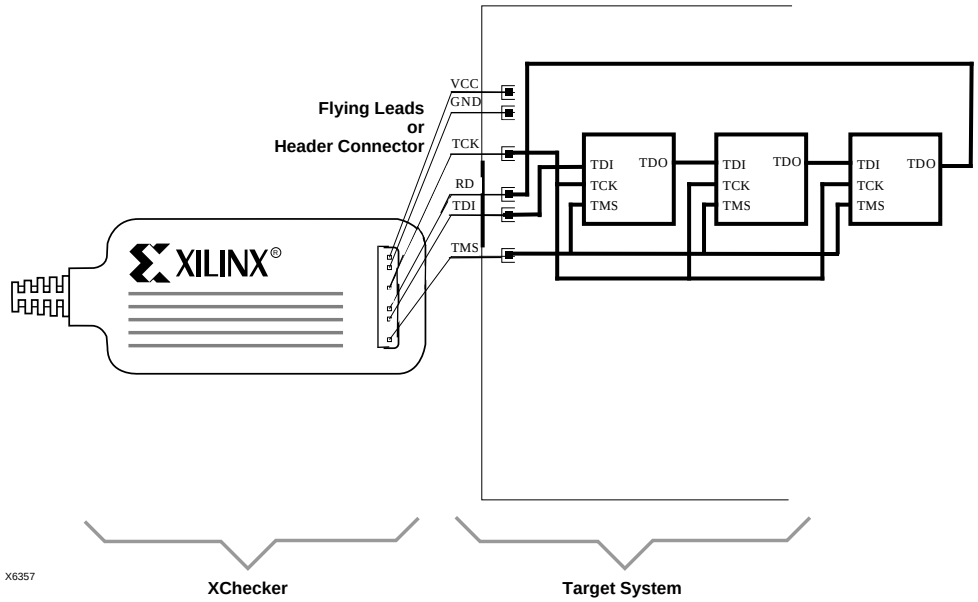


Figure 2-3 XChecker Connections to JTAG Boundary-scan TAP

Parallel Download Cable

The Parallel Download Cable consists of a cable assembly containing logic to protect your PC‘s parallel port and a set of headers to connect to your target system.

Using the Parallel Download Cable requires a PC equipped with an AT compatible parallel port interface with a DB25 standard printer connector. Figure 2-4 shows the Parallel Download Cable.

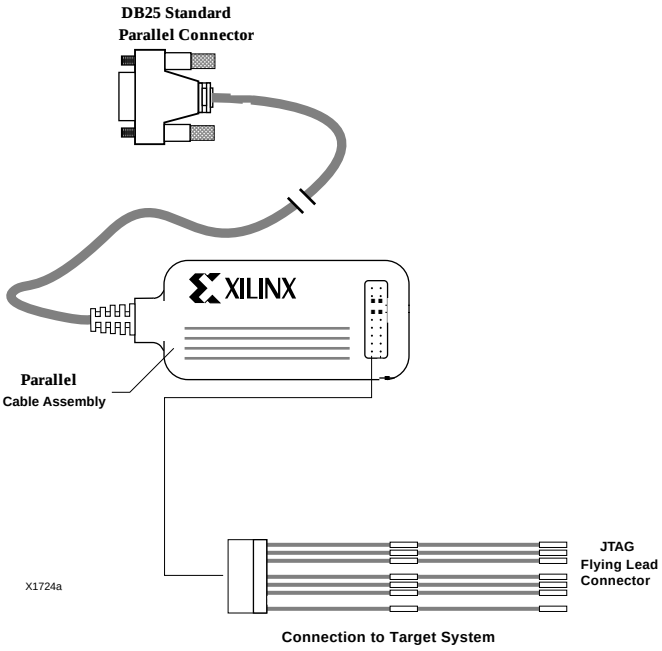


Figure 2-4 Parallel Download Cable and Accessories

The cable assembly contains logic designed to electrically isolate the target system from the parallel port of your PC host system.

The parallel download cable can be used with a single EPLD or several connected in a boundary-scan chain to download and read-back configuration data.

The transmission speed of the Parallel Download Cable is determined solely by the speed at which the host PC can transmit data through its parallel port interface.

Connecting the Parallel Download Cable

On PCs you can connect the parallel cable to your system's parallel printer port. The EZTag software will automatically identify the cable when correctly connected to your PC. If you choose to, you may also select this connection manually. To set up a parallel port manually:

Cable → Parallel Port → [*LPTn*]

Connection to Your Target System

You need appropriate pins on the target system for connecting the target system board to the header connection on the cable. These connectors must be standard 0.025–inch square male pins that have dedicated traces to the target system control pins. You can connect to these pins with a header connector or a flying lead connector.

Note: The parallel cable draws its power from the target system through V_{CC} and GND. Therefore, power to cable, as well as to the target system, must be stable. Do not connect any signals before connecting V_{CC} and ground.

Header Connector

When you layout the printed circuit board for use with JTAG in-system programming and testing, a few adjustments will make the process of connecting and downloading easier.

- Provide header pins on your printed circuit board for VCC, GND, TCK, TDI, TMS and TDO.

The header connector is a standard 9-pin (6 signals, 3 keys) header connector that fits 0.025″ square male pins. The pin order is listed in Table 2-3. These header connectors are keyed to assure proper orientation to the cable assembly.

Flying Lead Connectors

The flying lead connector is two flying lead header connectors with eight standard individual female connectors on one end that fit onto 0.025″ square male pins. Each lead is labeled to identify the proper pin connection.

Cable Connections

Connections between the cable assembly and the target system use 6 leads. For connection to JTAG boundary-scan systems you need to ensure that the VCC, GND, TDI, TCK, TMS and TDO pins are connected.

Once installed properly, the connectors provide power to the cable, allow download and readback of configuration data, and provide for logic probe of device pins.

Table 2-3 describes the pin connections to the target circuit board.

Table 2-3 XChecker Cable Connections and Definitions

Name	Function	Connections
VCC	Power – Supplies V_{CC} (5 V, 100 mA, typically) to the cable.	To target system V_{CC}
GND	Ground – Supplies ground reference to the cable.	To target system ground
TDO	Read Data – Read back data from the target system is read at this pin.	Connect to system TDO pin.
TDI	Test Data In – this signal is used to transmit serial test instructions and data.	Connect to system TDI pin.
TCK	Test Clock – this clock drives the test logic for all devices on boundary-scan chain.	Connect to system TCK pin.

Name	Function	Connections
TMS	Test Mode Select – this signal is decoded by the TAP controller to control test operations.	Connect to system TMS pin.

Connecting for System Operation

Connect the parallel cable to the host system and your target system as shown in Figure 2-5.

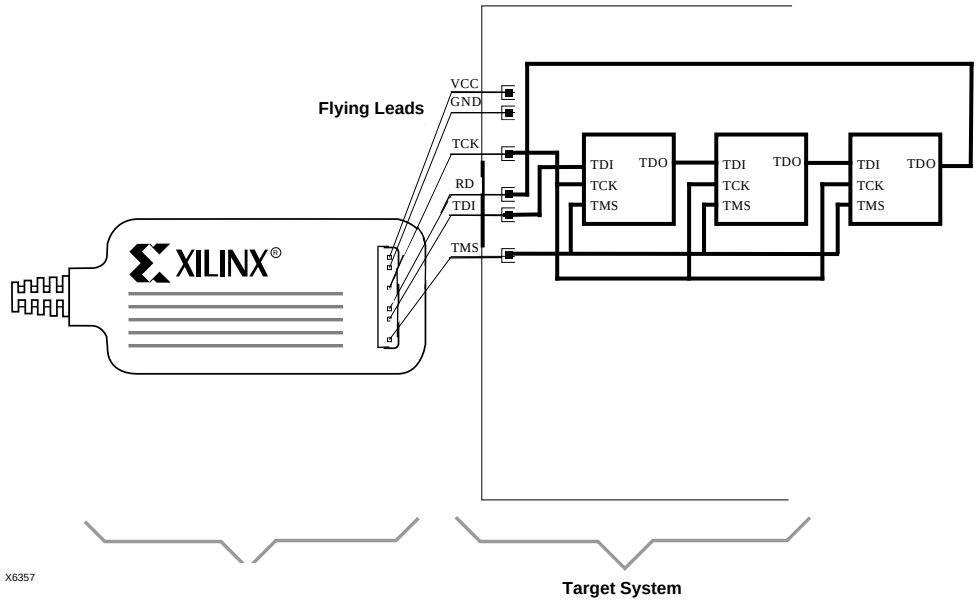


Figure 2-5 Parallel Download Cable Connection to JTAG Boundary-scan TAP

Chapter 3

In-System Tutorial for PCs

Introduction

This chapter will take you through the basic steps involved in programming an XC9500 family device in-system from a PC. EZTag supports XC9500 family devices, including:

- XC9536
- XC95108
- XC95216

Cable Setup

To setup your system to download configurations in-system you must first connect the EZTAG parallel download cable as follows:

1. Connect the EZTAG Download Cable to the parallel printer port of your PC. This cable contains drivers to buffer the signals as they are driven into the system.
2. Attach the flying leads to the EZTAG Download Cable (see Figure 3.1). Leads attach to the inside row. The connector is designed to fit only the correct way.
3. The power for the drivers is derived from the target system. Connect the cable's VCC and GND wires to the corresponding signals on the target board.
4. Next connect the JTAG TAP inputs. Connect TCK, TDI, TMS and TDO to the target board. TRST is not supported by the XC9500 EZTAG Download Cable. If any of your JTAG parts have a TRST pin, it should be connected to VCC.

Warning: Cable protection ensures that the parallel port cannot be damaged through normal cable operation. For increased safety, please power up the PC before the target system is powered up.

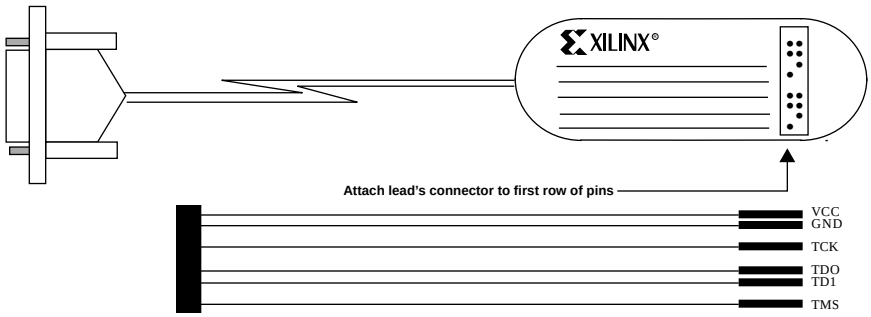


Figure 3-1 EZTAG Cable and Leads

Selecting a Port for the Cable

You may select a serial or parallel port for your cable from the EZTag Interface. To set up a parallel port:

Cable → Parallel Port → [*LPTn*]

To set up a serial port:

Cable → Serial Port → [*Comn*]

Device Chain

The device chain U1, U2, U3, ... U*n* is a serial chain where U1 is the first device TDI enters and U*n* is the last device. U*n* must deliver the TDO signal back to the cable. TMS and TCK signals enter all devices in parallel.

When you set up the EZTAG device chain you must list the devices in the chain in order even if they are in BYPASS mode. List every device in the chain whether they are Xilinx devices or some other third party device.

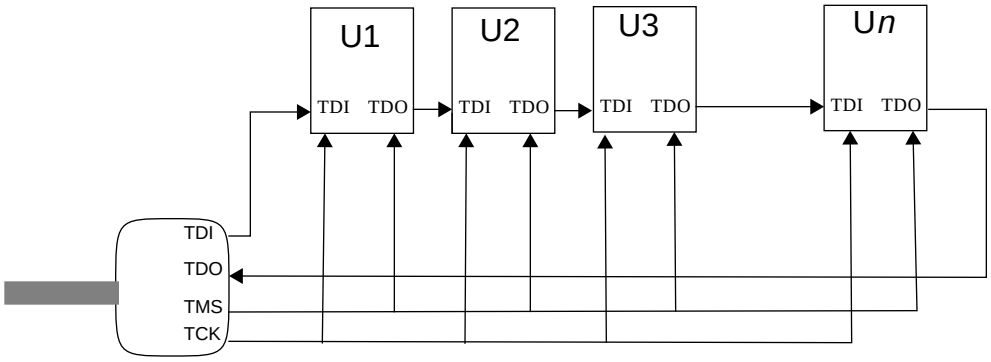


Figure 3-2 Device Chain

Configuring a Device In-System

If you have created programming files (**<*filename*>.jed**) and are ready to download them to XC9500 devices in-system through the JTAG chain, proceed as follows:

1. Make sure the cable is attached properly and the target board is turned on.

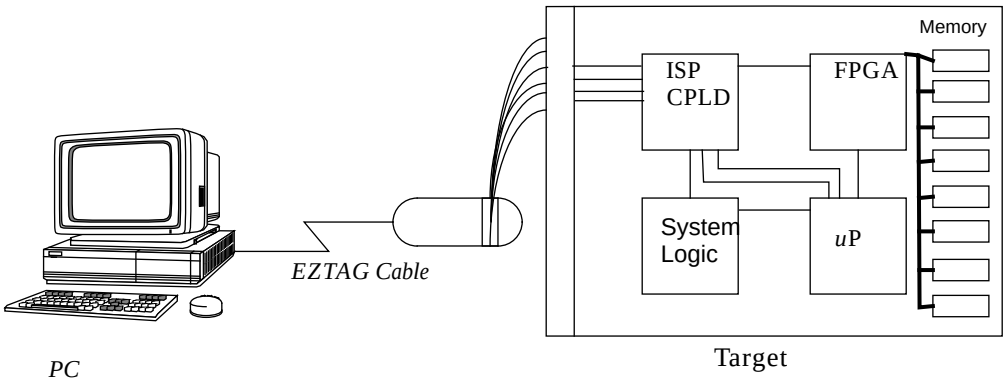


Figure 3-3 JTAG Connections

2. Make sure that the BSDL files for non-XC9500 devices are stored along the XACT search path.

3. Invoke the EZTAG-JTAG Download Software menu by double-clicking the EZTAG-JTAG Download Software icon.

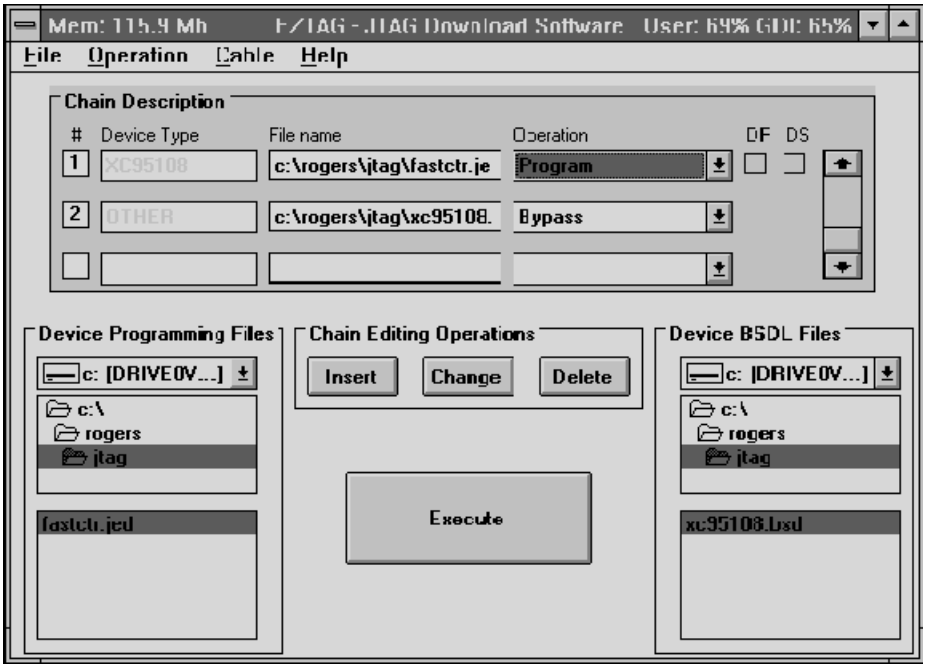


Figure 3-4 EZTAG Menu with U1 Set for Programming and U2 Set in Bypass Mode.

4. Select the operations desired for each XC95108 part.

   a) Under **`Device Programming Files`** find the disk and directory containing the files you want to use.

   b) Double-click on the file name. The **`Device Type`** and **`File name`** should appear in the **`Chain Description`**.

   c) Click once on the **`Operation`** down arrow to select the type of operation you want to select. Select **`Program`** (this is the default).

5. Select **`Bypass`** for each non-XC-9500 part. These parts will appear under **`Device BSDL Files`**. Double-click on the file for each part so that it appears in the **`Chain Description.`** Note that **`Bypass`** is selected as the default **`Operation`** of each foreign

part. **Bypass** is the only supported mode of **Operation** for non-FastFLASH parts.

Note: The **Chain Description** has a scroll bar to the right of it that will allow you to scroll around in the chain. If your chain has more than three entries, this bar is necessary to view them all.

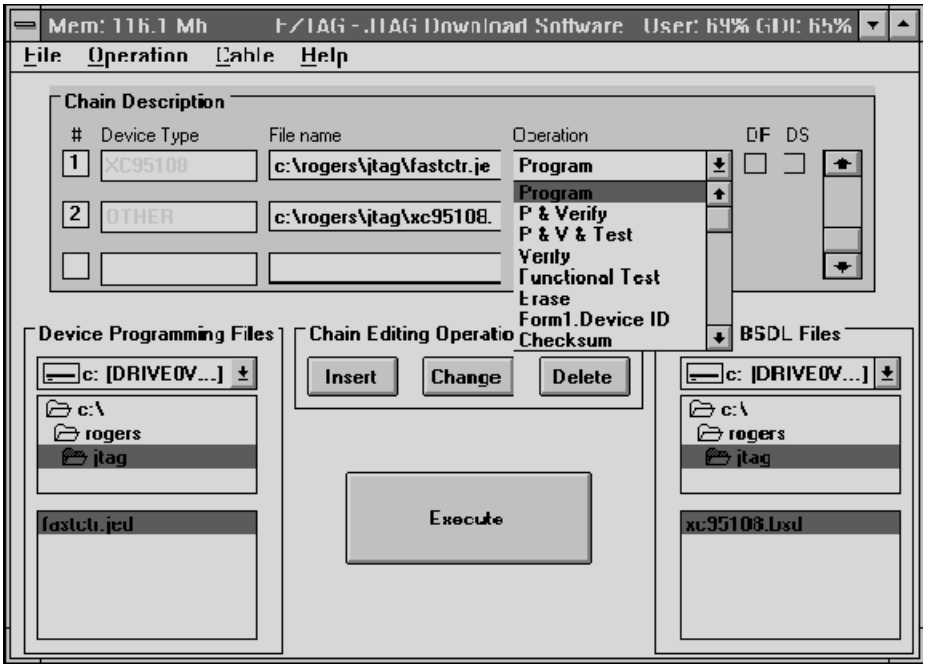


Figure 3-5 EZTAG Menu showing Operation Options

6. Select the **Execute** button. Download will begin. When Execute works properly a DOS window appears and delivers processing messages. When processing has completed, a message log appears and lets you know the status and results of the execution. If any errors occur, they will be listed in the message log.

Note: It will take between thirty and sixty seconds to download to a part using the Silicon Evaluation Release software.

Modifying a Chain

The **Chain Editing Operations** are located on the EZTAG menu. They softkeys listed as **Insert**, **Change**, and **Delete**. In

addition, if you want to start over, you can use **File → Clear Chain**. This will clear all entries in the **Chain Description**.

Insert

The insert softkey allows you to insert a new device into the chain. When you insert, the device number selected and all subsequent devices are moved down one in the chain. To insert a device, follow these instructions:

1. With the mouse, click once on the number (**#**) in the **Chain Description**. The number column is found to the left of the **Device Type**. The **#**, **Device Type** and **File Name** on that row should be highlighted, indicating that this is the selected element for this operation.
2. Click the **Insert** softkey once.
3. Select a file to enter into the chain and double-click. The file should enter at the highlighted line and push all files at and below the highlighted line down the chain by one. The highlighting disappears and the element number is no longer selected.

Change

The change softkey changes the entry on a Chain Description line, deleting the entry at the line and replacing it with a selected file. To change a device:

1. With the mouse, click once on the number (**#**) in the **Chain Description**. The number column is found to the left of the **Device Type**. The **#**, **Device Type** and **File Name** on that row should be highlighted.
2. Click the **Change** softkey once.
3. Select a file to enter into the chain and double-click. The device should enter at the highlighted line and replace the file that was on the line. The highlighting disappears and the element number is no longer selected.

Delete

The delete softkey deletes an entry in the **Chain Description**. All items below that line move up one entry in the chain.

1. With the mouse, click once on the number (**#**) in the **Chain Description**. The number column is found to the left of the **Device Type**. The **#**, **Device Type** and **File Name** on that row should be highlighted.
2. Click the **Delete** softkey once. The highlighted line will be deleted and all items in the chain below that line will move up one place in the chain.

Note: When no chain editing operation is active, selected files (by double-clicking) are added to the end of the chain.

Saving a Chain

To save an EZTAG chain description for later use, create a Chain Description File (**.cdf**) using:

File → Save

A screen titled **CDF File Operation** will appear. This screen will allow you to select a directory and path to place the file in. You can also name the file, but you should retain the **.cdf** file extension.

To name your file, use the mouse to highlight the asterisk (*) on the **File Name** line, then type in the name you want and click once on **OK**.

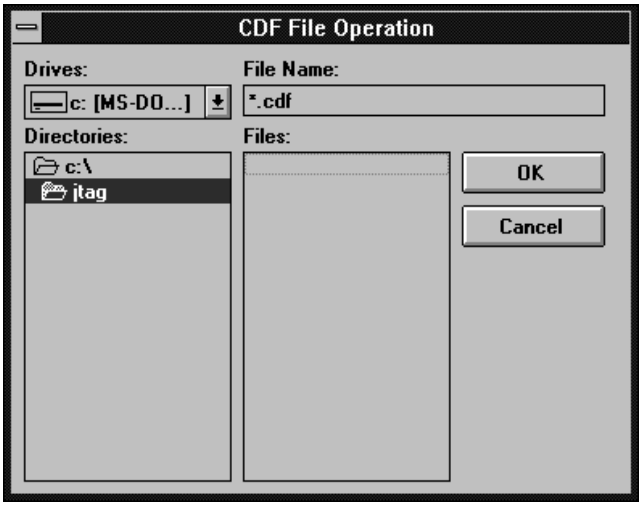


Figure 3-6 Saving a File

Saving a Modified Chain

If you use a previously saved chain description as a template for creating another chain description, or want to modify a description and save it under another name, use the **Save As** command.

File → Save As

This will bring up the **CDF File Operation** menu. To name your file, use the mouse to highlight the asterisk (*) on the **File Name** line, then type in the name you want and click once on **OK**.

Data Security Selection

Any Xilinx device selected for programming can be secured with the **Data Security** (**DS**) or **Data Protection** (**DP**) or both.

When enabled, **Data Security** disables reading the programmed contents of a device (the **man.id** and signature remain readable).

Data Protection allows only the reading of the programmed data. The device contents cannot be altered.

When both **Data Security** and **Data Protection** are enabled, the device can be neither read nor re-programmed.

To enable **Data Security** or **Data Protection** or both:

1. Find the **DS** and **DP** checkboxes in the Chain Description. They should only appear in rows in which programming is involved (**Program**, **P & Verify**, and **P & V & Test**).
2. Enter a check in either or both boxes by clicking once on the box. If you change your mind about security, click once on the box again. The check will disappear.

Chapter 4

EZTag with Workstations

This chapter gives specific information about using EZTag in a workstation environment to perform JTAG operations. You can use EZTag to download, read back, verify design configuration data for any device, and to probe internal logic states of an EPLD design.

EZTag software supports the XC9500 family of Xilinx EPLD devices, including:

- XC9536
- XC95108
- XC95216

EZTag software support the following capabilities.

- EZTag allows you to download a design to the EPLD on the target system.
- EZTag can verify EPLD configuration by comparing it to the original JEDEC programming file after configuring an EPLD.
- You can program multiple EPLDs connected on a boundary-scan chain.
- You can apply test vectors from a JEDEC file through the boundary-scan TAP to EPLDs using the INTEST instruction.

Using the EZTag Software

This section describes the EZTag files and commands.

EZTag Files

You must become familiar with the following files, which are used by the EZTag software.

design.jed

The *design*.jed file contains the configuration information for the target design in JEDEC 3-C standard formats. The file is generated by the fitter software. This file may optionally contain functional test vectors to do functional verification of XC9500 devices.

eztag.pro

The eztag.pro file contains the default values for all EZTag options: part, design, baud, and port. These option values are updated at the end of every EZTag session. For EZTag to recognize an xchecker.pro file, it must be located in the same directory in which you started the EZTag software.

batch_file.cmd

The batch files are text files used to execute commands in the batch mode, and the extension ".cmd" is required.

device.bsd

The bsd files contain Boundary Scan Description Language (BSDL) specifications of the operation of the boundary-scan logic of a given device. For any non-XC9500 device in your boundary-scan chain, you are required to supply this file.

Invoking EZTag

You can start EZTag using any of the following methods:

- Command line entry from the system shell. Only download and verify are supported. You cannot interactively probe the internal logic. Command line entry can only be used when there is exactly one device in the boundary-scan chain.
- Interactive commands from the system shell. This mode offers additional commands for download and readback and also allows you to probe the internal logic states of the target system device.

Downloading

You can download a design after connecting the XChecker cable to the host system and target system. To download a design, enter the

following command at the operating system prompt.

eztag *design_name*

When you do not specify any options, the EZTag software selects the port where the cable is connected and sets the baud rate to the maximum allowed by the platform. You can modify the communication port and baud rate by changing the appropriate settings in the xchecker.pro file.

1. To download in an interactive mode, enter the following command at the system prompt.

   eztag

   You see the following message on the screen:

   ```
   Xilinx (R) EZTAG XC9500-Pre-Release-V2.0.a-JTAG
   Boundary-Scan Download
   Copyright (C) Xilinx, Inc. 1991-1995. All rights
   reserved.
   Cable ID type is 'XCHECKER'
   Cable is connected to '/dev/ttya'
   Baud rate is 38400
   ```

2. To specify the number, type, names and order of devices in the boundary-scan chain:

   part *part_type*:*design_name*

3. To program the a design, enter this command string:

   program *design_name*

Verifying

After you have properly configured a device, you can verify its configuration and compare it to your original design.

In most applications, verification is not needed, but this feature can be helpful with designs that experience extremely unstable or noisy V_{CC} conditions.

To download and verify a design, enter the following command string at the operating system prompt.

eztag –v *design_name*

Specifying the –v option causes the XChecker cable to download the

file *design*.jed to the target system and initiate a readback immediately after the configuration data has been downloaded.

To execute a readback after the device has been in operation, use the interactive commands, as follows:

`eztag`

This command invokes the interactive mode, and the `EZTAG ?` prompt appears.

`part` *part_type:design_name*

The part commands identifies the number the number, type, name and order of devices in the boundary-scan chain. In this case there is one device only. Then to program the device, enter:

`eztag ? program` *design_name*

The program command downloads *design*.jed to the target device. If you want a readback after the target device is in operation, you can execute the Verify command.

`eztag ? verify` *design_name*

This command initiates a readback, and compares the data to the *design*.jed file.

You may also execute the program and verify operations in one step by typing:

`eztag ? program -v` *design_name*

Command-Line Options

This section describes the EZTag command-line options. The data files are configuration bitstream files in JEDEC format. When you do not specify any options or data files, the XChecker system defaults to the interactive mode. If you do not specify any options but you do specify a data file, a default set of options specified in the xchecker.pro file sets the port.

The command-line syntax is as follows:

`eztag options` *datafile*

Note: You can abbreviate all options to the minimum number of distinctive characters in the option name.

Commands and options are not case-sensitive.

–batch Batch Mode Operation

Syntax –batch *bat_file*.cmd

Abbreviation b

The Batch option executes commands in batch mode. The *bat_file* must have a ".cmd" extension and contain valid EZTag commands, including interactive commands. You can add comments to files by using the # symbol, either on the command line or on a new line.

–h The Help Option

Syntax –help

Abbreviation h

The Help option displays command line usage information.

–pa Specify Part Type

Syntax –part *parttype:design_name*

Abbreviation pa

The Specify Part Type option defines the part to be used such as XC95108 and XC95216. This option is always required and specifies the number, type, name and order of devices in the boundary-scan chain. The parts are specified in order from the closest to TDI to the closest TDO.

–po Specify Port Name

Syntax –port *portname*

Abbreviation po

The Specify Port Name option identifies the port connection for the XChecker cable. If you do not specify this option, the default option AUTO, searches for the cable connected to any port, parallel or serial. Valid ports for supported platforms are listed in Table 4-1.

Table 4-1 Valid Ports for the XChecker Cable

Platform	Communication Ports			
IBM PC	com1	com2	lpt1*	lpt2*
NEC PC	com1			
Sun	/dev/ttya**	/dev/ttyb**		
HP700	/dev/tty00	/dev/tty01		

*Use with the parallel download cable only.

**ttya and ttyb must be readable and writable to ensure a proper connection.

–v Verify Download and Readback

Syntax –v

Abbreviation v

The Verify Download and Readback option executes a download and readback of the current EPLD design for verification. EZTag reads the configuration from the EPLD and compares it to the original bitstream. If you do not specify this option, readback is not executed after configuration.

Interactive Mode Commands

This section describes the EZTag interactive mode commands. To use the interactive mode commands, you enter **`eztag`** at the system prompt.

Note: You can abbreviate the commands using the least number of distinctive characters, as with the command line options, but you must use at least two characters. You can repeat the previous command using either an equal sign, "=," or an exclamation point, "!."

Batch — Execute in Batch Mode

Syntax batch *bat_file*.cmd

Abbreviation bat

The Batch command executes commands in a batch mode. The *bat_file* must have a ".cmd" extension and contain valid EZTag commands. Use the pound sign, "#" to precede comment lines in the batch file.

Examples

The following examples show two methods of using the Batch command from the EZTag prompt:

```
batch bat_file.cmd
< bat_file.cmd
```

Baud — Specify Baud Rate

Syntax baud *baud_rate*

Abbreviation bau

The Baud command specifies a communication baud rate. At initialization, the fastest baud rate for your host system is automatically selected. Table 4-2 lists the valid baud rates.

Table 4-2 Valid Baud Rates

Platform	Baud Rate			
	9600	19200	38400	115200
IBM PC	X	X	X	X
NEC PC	X			
Sun	X	X	X	
HP 700	X	X	X	

X indicates valid baud rate

Dump

Syntax dump *part_name* -j *file_name*

Abbreviation

The dump command will read the contents of a part and create a

JEDEC file with the results. The file created will default to part_name.jed. Optionally, you may specify your own name using the -j flag. The *part_name* must have been specified with the **part** command.

Erase

Syntax erase *part_name*

Abbreviation

This command erases the programmed contents of the specified part. The *part_name* must have been specified with the **part** command.

Exit — Terminate Session

Syntax exit

Abbreviation exi

The Exit command terminates the current EZTag session, asks you whether to save current program options in the xchecker.pro file, and returns you to the system shell.

Functest

Syntax functest *part_name* [-j *file_name*]

Abbreviation

The functest command will run the functional vectors in the associated JEDEC file (*file_name*) on the specified device (*part_name*) using the intest command. If the *part_name* is the same as the JEDEC *file_name*, then the *file_name* does not need to be specified. The *part_name* must have been specified with the **part** command.

Help — Online Help

Syntax help *topic*

Abbreviation he

The Help command displays online help for the topic requested in 24-line segments. Enter **y** to scroll forward to the next 24 lines. Enter **n** to exit Help.

Log — Send Screen Display to File

Syntax log –out *file_name string*

Abbreviation log

The Log command sends the screen output to the *file_name* file. Use this command to capture the output of a Readback or a Show command.

There is one option for the Log command.

–out

The –out option closes any previous log file, opens a new one, and places the string at the beginning of the file.

There is one variable for the Log command:

string

Use the variable *string* to insert your comments into the log file, which normally only captures the screen display.

Part — Specify Device Chain

Syntax part *device_type:part_name device_type:part_name ...*

Abbreviation pa

This command must be executed first. It describes the devices in the chain to the software. The *device_type* is used to find the BSDL file associated with each part. BSDL files must be named *device_type*.bsd. The *part_name* is an arbitrary name to associate with the device instance in the chain. It will usually be the proper name (the file name without the extension) of the JEDEC file associated with the device at that location in the boundary-scan chain, although it could be anything. The boundary scan chain order must start with the closest device to TDI, and proceed in order through the chain until it reaches the last device, which is closest to TDO. When multiple "**`part`**" commands are issued, the information associated with the very last is maintained.

Partinfo

Syntax partinfo -id -signature -checksum *part_name*

Abbreviation

The partinfo command returns the manufacturer's identification (id), the user signature (-signature) or the device checksum (-checksum) for a particular *part_name*. Any or all of the three switches may be specified in a single command. The *part_name* must have been specified in the **`part`** command.

Port — Specify Download/Readback Port

Syntax port *portname*

Abbreviation po

The port command specifies the download / readback port. Table 3-5 lists the valid entries; the ports listed in bold face are the defaults. If the port is defined as Auto, all ports are scanned to search for a cable.

Table 4-3 Valid Ports for the XChecker Cable

Platform	Communication Ports			
IBM PC	com1	com2	lpt1*	lpt2*
NEC PC	com1			
Sun	/dev/ttya**	/dev/ttyb**		
HP700	/dev/tty00	/dev/tty01		

*Use with the parallel download cable only (no readback).

**ttya and ttyb must be readable and writable to ensure a proper connection.

Program

Syntax program [-v] [-t] [-s] [-p] *part_name* [-j *file_name*]

Abbreviation

This command programs the specified *part*. If the *part_name* is the same as the JEDEC *file_name*, then the *file_name* does not need to be specified. The *part_name* must have been set in the **`part`** command. There are four options that may be specified (individually or together):

`-v` after programming the device reads back the contents and verifies that they agree with the associated JEDEC file.

`-t` executes a functional test after programming using the vectors

contained in the associated JEDEC file.

-s sets data security in the device. This disables readback of the device's programmed contents. The device must be erased to reprogram it.

-p sets data protect in the device. This disables over-write of the device's programmed contents. The device cannot be erased or re-programmed.

Quit — Terminate Session

Syntax quit

Abbreviation qu

The Quit command terminates the current EZTag session and asks you whether to save current program options in the xchecker.pro file.

Reset — Reset Target LCA/Cable

Syntax reset [–cable]

Abbreviation res

The Reset command resets the boundary-scan TAP state machines or the XChecker cable. The default is to reset the boundary-scan TAP state machines.

There is one option for the Reset command.

–cable

The –cable option reprograms the XChecker cable's internal FPGA. It re-initializes the cable, including setting the correct baud rate. This option is useful in the event of power glitches that could affect proper cable operation.

With this option, you could remove power from the target system, then restore power, while running EZTag; the Reset command re-initializes the cable to the proper settings.

Save — Save Option Settings

Syntax save

Abbreviation sa

The Save command saves the settings of four interactive command results in the xchecker.pro file; baud rate (Baud command), design name (Load command), device type (Parttype command) and port name (Port command).

At initialization, EZTag reads the xchecker.pro file to set up the defaults for the current session. This file must be in the current directory or in the XACT environment search path. EZTag updates the profile information at the end of every session. The xchecker.pro file is created when you exit from your first EZTag session.

Settings — Display Settings

Syntax settings

Abbreviation se

The Settings command provides a listing of the following information; the port name, the baud rate, the type of cable, the design name, the part type and package type, the clock source, and hardware trigger status. It also lists the number of clocks for the first and subsequent snapshots, the number of signals defined in the probe list, and the number of signals defined in the display list.

Sys —Temporarily Exit to Operating System

Syntax sys

Abbreviation none

The Sys command allows you to temporarily exit from EZTag to the operating system prompt. Enter **`exit`** to return to EZTag.

Verify — Verify Target EPLD Bitstream

Syntax verify *part_name* [-j *file_name*]

Abbreviation ve

This command reads back the configuration registers of the specified part and compares its contents against the JEDEC file. If the *part_name* is the same as the JEDEC *file_name*, the *file_name* does not need to be specified.

Troubleshooting Guide

This section is a simple guide to understanding the more common issues you might encounter when configuring EPLDs with EZTag. These issues are likely to fall into three groups; communication, improper connections, and improper or unstable V_{CC}.

- Communication

  This section describes several issues that involve the integrity of the bitstream that EZTag transmits to the target EPLDs, and the correct connection of the boundary-scan chain.

- Improper Connections

  This section involves assigning configuration pins to invalid signals or voltage levels.

- Improper or Unstable V_{CC}

  This section describes several causes of incorrect configuration sequences and incorrect responses from the target system.

Communication

Observing the following guidelines should minimize the communication difficulties that can occur between the XChecker hardware and the target system.

Do not attach extension cables to the target system side of the XChecker cable; this can compromise configuration data integrity and cause checksum errors.

Attach the XChecker cable configuration leads firmly to the target system.

After connecting the target system, specify the chain configuration using the "**`part`**" command. Then use the "**`partinfo -id`** *part_name*" command to read the IDCODE from each part in the system. This will verify the integrity of the boundary-scan chain.

Use the verify feature to assure integrity of the configuration data. You can do this from the command line with the –v option or in the interactive mode by specifying the verify command.

Improper Connections

Always make sure that XChecker leads are connected properly.

Note: Connecting the XChecker leads to the wrong signal will cause permanent damage to XChecker internal hardware. You must connect V_{CC} to +5 V and gnd to ground.

For workstations, you must have read and write permissions to the port to which you connect the XChecker cable. EZTag might issue a message stating that the cable is not connected to port tty*x*. When you see this message, follow the check list below:

- The board must have the power on, since the XChecker Cable uses power from the board.
- Check the device driver using the following command string:

  `ls –l /dev/ttya /dev/ttyb`

The result should be the following:

```
crw-rw-rw-  1 root12,0 month date time /dev/ttya
crw-rw-rw-  1 root12,1 month date time /dev/ttyb
```

- Reconnect the XChecker cable to another valid port.
- Read the /etc/ttyab file. There should be two lines, as follows:

```
ttya   ‘‘/usr/etc/getty std.9600’’  unknown   off
local secure
ttyb   ‘‘/usr/etc/getty std.9600’’  unknown   off
local secure
```

If you use a port to connect a modem or a remote login, you cannot use that port. The port must be on. Consult your System Administrator if the information the /etc/ttyab file if different than what is listed in the aforementioned list.

Improper or Unstable V_{CC}

Never connect the control signals to XChecker before V_{CC} and ground. Xilinx recommends the following sequence:

1. Turn off power to the target system.
2. Connect V_{CC}, ground, and then the signal leads,

3. Turn on power to the target system.

Warning: As with any CMOS device, the input/output pins of the internal FPGA should always be at a lower or equal potential than the rail voltage to avoid internal damage.

Make sure V_{CC} rises to a stable level within 10ms. Stable V_{CC} should be between 4.75 V and 5.25 V.

In the event of power glitches, use the interactive Reset command with the –c option (Cable option) to reconfigure the XChecker internal FPGA and use the interactive program command to reconfigure the target EPLD.

Appendix A

Error Messages

Introduction

This section describes the error messages that EZTag may generate. Following each error message, there is a suggested workaround.

Error Messages

`Error 001: Command file` *bat file.cmd* `is not found.`

Make sure that the command file you specified is in the current directory or the environment search path. Make sure that the command file has the ".cmd" extension

`Error 002: Internal Error – Command table syntax error Cmd=valid_command.`

This is an internal program error that normally should not occur. Try entering the command sequence again. If the error persists, try reinstalling your EZTag software. If the error reappears, call Xilinx Technical Support. Be prepared to duplicate the error and reference specific files or examples.

`Error 003: Diagnostic is not supported on this cable.`

The Diagnostics command, as well as other readback and verification commands are only supported for the XChecker cable.

`Error 010: Cannot open output file` *file_name.*

Check available disk space. Current directory or file must have write permission.

`Error 011: Cannot create output file` *file_name.*

Check available disk space. Current directory or file must have write permission.

`Error 012: Cannot open input file` *file_name.*

Make sure the *file_name* file exists in your working directory or in the environment search path. Current directory or file must have write permission.

`Error 013: File` *file_name* `is not found.`

The *file_name* file does not exist in the current directory or search path. Make sure that the *file_name* file exists in your working directory or in the environment search path.

`Error 021: Help file` *eztag.hlp* `is not accessible`

Make sure that the XACT environment variable points to the XACT directory in the PC (the XACT directory in the PC is equivalent to the "Installation Directory" on the workstations). Also make sure that eztag.hlp is in the directory XACT\MSG. If you cannot find eztag.hlp in the XACT\MSG directory, you must reinstall the XChecker software.

`Error 022: No help for command` *command entered.*

Help is not available for the specified command. Refer to the Interactive Mode Commands section in Chapter 3 for help.

`Error 023: Cannot save configuration to` *file_name.pro.*

Check available disk space. Current directory or file must have write permission.

`Error 024: Invalid command at line` *line number.*

Check the file xchecker.pro in your current directory for illegal commands. Delete the xchecker.pro file. EZTag creates a new profile when you exit from the session.

`Error 030: Ambiguous command.`

Enter the minimum unique characters that identify the command or enter complete commands with no abbreviations.

`Error 031: Invalid command.`

The command you entered is illegal. Refer to the Interactive Mode Commands section in Chapter 3 for help.

`Error 032: Invalid number of arguments.`

Refer to the Interactive Mode Commands section in Chapter 3 for help.

`Error 033: Invalid option` *selected option.*

Refer to the Command-Line Options and the Interactive Mode Commands sections in Chapter 3 for help.

`Error 034: Invalid value given to` *parameter.*

Refer to the Command-Line Options and the Interactive Mode Commands sections in Chapter 3 for help.

`Error 035: Value is required for` *command entered.*

Refer to the Interactive Mode Commands section in Chapter 3 for help.

`Error 050: System Error Messages`

System error codes are usually a string of messages generated by your operating system.

`Error 051: System file error code.`

System error codes are usually a string of messages generated by your operating system.

`Error 101: Cable is not initialized.`

Reissue the Reset command with the –c option, or cycle power to the XChecker cable and then issue the Reset command with the –c option. See the Improper or Unstable V_{CC} section in Chapter 3.

`Error 102: Cable is not located.`

No cable has been recognized at any port. Make sure there is power to your board and to the XChecker cable. If you are using the test fixture, you must connect V_{CC} and ground to it. The XChecker cable draws power from your target system, not from your host computer. Also make sure the RS-232 connector is firmly attached.

`Error 103: Invalid port name.`

Refer to the XChecker Hardware section in Chapter 3 for help.

`Error 104: Invalid baud specified.`

Refer to the XChecker Hardware section in Chapter 3 for help.

`Error 105: Cable is not reset.`

Cycle power to the cable. Use the Reset command with the –c option.

`Error 107: Communication line is broken.`

Run the Reset command with the –c option. Also make sure there is power to your board and to the XChecker cable. Check all power and port connections.

`Error 108: Communication checksum error.`

Check for induced noise in your target system or from your target system into the XChecker connections. Do not use cable extensions. The XChecker cable length is tested to produce minimal noise levels. Remember that a logic High must be 80-100% of V_{CC} and a Logic Low must be 0-25% of V_{CC}.

`Error 109: Cable has no power.`

Make sure there is power from your target system to the XChecker cable. The cable draws power from an external source, not from the host computer.

`Error 110: Communication time-out.`

EZTag has not received an expected signal; for example, a system trigger to initiate readback or data coming from readback. Make sure that the selected options for trigger and readback are what you intended. Check all connections.

`Error 120: Cannot communicate to the cable.`

Run the Reset command with the –c option. Also, ensure that there is power to your board and to the XChecker cable. Check all connections. Make sure the RS-232 connector is firmly attached.

`Error 121: Cable datafile` *file_name* `is empty.`

Run the Reset command with the –c option. Make sure that the XACT environment variable points to the XACT directory on the PC. (On workstations, the XACT directory is equivalent to the "Xilinx_Directory" referenced in the installation notes).

`Error 122: Cable datafile` *file_name* `has invalid format.`

On the workstations, the XACT directory is equivalent to the "Xilinx_Directory" referenced in the installation notes.

`Error 122: Can't open cable datafile` *file_name.*

On the workstations the XACT directory is equivalent to the "Xilinx_Directory" referenced in the installation notes.

`Error 123: No XChecker cable is connected to the port` *portname.*

Ensure that there is power to your board and to the XChecker cable. You must connect V_{CC} and ground to the test fixture, if you are using it. The cable draws power from your target system, not from the host computer. Ensure that the RS-232 connector is firmly attached.

`Error 124: No XChecker cable is connected to the system.`

Ensure that there is power to your board and to the cable. You must connect V_{CC} and ground to the test fixture, if you are using it. XChecker draws power from your target system, not from the host computer. Ensure that the RS-232 connector is firmly attached.

`Error 125: Fail reading cable status.`

Try using the Reset command with the cable option. Ensure that there is power to your board and to the cable. Check all connections.

`Error 126: Unsupported command for this cable.`

See the Interactive Mode Commands section for valid with the XChecker cable. If you are using the previous parallel or serial download cables, you can only use the Load command to download.

`Error 128: Read only` *number of bits received.*

Check all connections. Check for noise that may be induced into your target system or from your target system into the XChecker connections. Do not use cable extensions. The XChecker cable length is tested to produce minimal noise levels. Remember that a logic High must be 80-100% of V_{CC} and a Logic Low must be 0-25% of V_{CC}.

`Error 130: Invalid baud rate. Current baud rate is` *baud rate.*

See Table 3-1 for the valid baud rates for your computer.

`Error 131: Missing baud rate. Current baud rate is` *baud rate.*

See the Interactive Mode Commands section in Chapter 3 for correct command usage.

`Error 134: Cannot communicate with port` *port name.*

Check this manual for supported ports. See the Port command.

`Error 135: Invalid port name` *port name.*

Refer to Table 3-5 for supported ports. See the Port command.

`Error 141: Datafile` *file_name* `is empty.`

The specified datafile is either empty or contains invalid data. EZTag supports only JEDEC 3-C format.

`Error 142: Datafile` *file_name* `is not found.`

The *file_name* file does not exits in the current directory or search path. Check your environment search path to make sure that it contains the directory where *file_name* is.

`Error 143: Can't open datafile` *file_name.*

The file *file_name* does not exits in the current directory or search path. Check your environment search path to make sure that it contains the directory where *file_name* is located.

`Error 220: No part type is defined.`

The part type specified in your design or by you (using the Part command) is invalid. Check the *The Programmable Logic Data Book* for valid part types and packages.

`Error 1001: Unable to execute erase command at address` *`string`* `of instance` *`string`*`.`

The specified device instance could not be erased. Check if data protect is enabled as this disables the erase functionality. Also check for the integrity of the cable connections.

`error 1003: Unable to program all addresses of instance` *`string`*`.`

The specified device instance could not be programmed. Check if data protect or data security is enabled as this disables the programming functionality. If data security is enabled, first issue an "erase" command then execute the program command. Also check for the integrity of the cable connections.

`error 1004: Verification of instance` *`string`* `against program file` *`string`* `failed.`

The specified device instance could not be verified. Check if data security is enabled as this disables the readback functionality. If this is not the case, check for the integrity of the cable connections. If error persists you may have a bad part and Xilinx customer service

should be contacted.

```
Error 1005: Unable to program address string of
instance string with data string.
```

The specified device instance could not be programmed. Check if data protect or data security is enabled as this disables the programming functionality. If data security is enabled, first issue an "erase" command then execute the program command. Also check for the integrity of the cable connections.

```
error 1006: Unable to verify address string of
instance string against data string.
```

The specified device instance could not be verified. Check if data security is enabled as this disables the readback functionality. If this is not the case, check for the integrity of the cable connections. If error persists you may have a bad part and Xilinx customer service should be contacted.

```
error 1011: Verification failed at address value of
instance string. Expected: value. Read: value.
```

The specified device instance could not be verified. Check if data security is enabled as this disables the readback functionality. If this is not the case, check for the integrity of the cable connections. If error persists you may have a bad part and Xilinx customer service should be contacted.

```
Error 1014: A description for a device named string
has not been supplied. Please make sure that a
BSDLdescription was loaded for this device.
```

```
Error 1015: A description for an instance named string
of any device has not been supplied. Please make sure
that a JTAG connection description was supplied for
this device.
```

Check that the specified part exists in the boundary-scan chain that you declared in your "part" command. A new "part" command will override the previously specified one. The current "part" database can be displayed by typing "part" followed by a carriage return.

```
Error 1017: The boundary scan chain instruction
register bit sequence is incorrect at bit value. This
corresponds to a scan chain break at or near part
string.
```

Verification of the integrity of the boundary-scan chain failed. Check

cable connections and “part” command specification The current “part” database can be displayed by typing “part” followed by a carriage return.

```
Error 1018: In a multi-part boundary scan chain, the
name of the particular boundary scan part instance on
which to operate must be specified. Please retry this
command with an instance name specified.
```

You must specify a particular instance upon which to operated. Respecify the command with that information. That is, specify “erase instanceName” and not “erase”.

```
Error 1021: Unable to execute erase command for
instance string
```

The specified device instance could not be erased. Check if data protect is enabled as this disables the erase functionality. Also check for the integrity of the cable connections.

```
Error 1022: Unable to execute functional test command
using vectors in JEDEC file string.

Error 1023: Functional test vectors failed for
instance string.

Error 1024: Functional test vector value failed for
instance string at pin number value. Expected output
value: value Actual output value: value
```

When running functional test using the INTEST instruction, the applied functional vectors mismatched the predicted values. This can be either a functional error in the design or an error in the vectors specified. This will also occur when vectors targetted for a different design are applied. Re-check the integrity of your design database information.

```
error 1025: Mismatched address values during
verification of instance string. Check JEDEC file and
cable connections.  Expected addressvalue: value.
Read: value.
```

The specified device instance could not be verified. Check if data security is enabled as this disables the readback functionality. If this is not the case, check for the integrity of the cable connections. If error persists you may have a bad part and Xilinx customer service should be contacted.

`Error 1026: Illegal IDCODE read from device identification register on instance` *`string`*`. IDCODE value:` *`string`*

The IDCODE read from the specified part does not conform to the 1149.1 standard. This is often the result of a bad cable connection. Check the integrity of the cable connection.

`Error 1028: Error reading data value from address` *`string`* `on device` *`string`* `while calculating checksum.`

`Error 1029: Data integrity errors while reading data values from device` *`string`* `will result in an incorrect checksum.`

While reading back data to calculate the checksum, errors occured. Check if data security is enabled as this disables the readback functionality. If this is not the case, check for the integrity of the cable connections. If error persists you may have a bad part and Xilinx customer service should be contacted.

`Error 1030: Unable to program data protect bit at address` *`string`* `on device string.`

`Error 1031: Programming failures when programming data protect bits of device string.`

`Error 1032: Unable to program data security bit at address string on device string.`

`Error 1033: Programming failures when programming data security bits of device string.`

The specified device instance could not be programmed. Check if data protect or data security is already enabled as this disables the programming functionality. If data security is enabled, first issue an "erase" command then execute the program command. Also check for the integrity of the cable connections.

`Error 1034: Error reading data value from address` *`string`* `on device` *`string`* `while generating JEDEC file.`

`Error 1035: Data integrity errors while reading data values from device string will result in an incorrect or incomplete JEDEC file.`

While reading back data to generate a JEDEC file, errors occured. Check if data security is enabled as this disables the readback functionality. If this is not the case, check for the integrity of the cable connections. If error persists you may have a bad part and

Xilinx customer service should be contacted.

```
Error 1036: Xchecker configuration file for boundary-
scan TAP driver not found. Check XACT path setting and
locate file named 'xckjtag.sys'.
```

When the Xchecker reconfiguration file xckjtag.sys is not found or loaded correctly the above messages are displayed. Check that your XACT path includes the release "data" directory and that the file "xckjtag.sys" exists in it. Also, check the integrity of the connections to the xchecker cable both at the serial port and to the target system.

```
Warning 1037: Data protection is enabled in instance
string (NOTE: device programming contents cannot be
altered).
```

This is the warning message issued when data protect is enabled. It is displayed with each operation addressing this device.

```
Warning 1038: Data security is enabled in instance
string (NOTE: device programming contents cannot be
read).
```

This is the warning message issued when data security is enabled. It is displayed with each operation addressing this device.

```
Error 1039: The device string is not a Xilinx part
(IDCODE: string)
```

```
Error 1040: The device string is not a XC9500 part
(IDCODE: string) Please verify the specification of
the order of the parts in the boundary-scan chain.
```

```
Error 1041: The device string is not an XC95108 part
(IDCODE: string). Please verify the specification of
the order of the parts in the boundary-scan chain.
```

```
Error 1042: The device string is not a currently
supported XC9500 part (IDCODE: string) Please verify
the specification of the order of the parts in the
boundary-scan chain.
```

These messages are displayed when the software identifies that the specified operation is targetting an improper device. Check that the specified part exists in the boundary-scan chain that you declared in your "**part**" command. A new "**part**" command will override the previously specified one. The current "part" database can be displayed by typing "**part**" followed by a carriage return.

```
Error 1043: The JEDEC file string is for a device of
type string. The specified part string is actually a
string device.  Please re-generate your JEDEC file.

Error 1044: The specified part string is of type
string for which JEDEC files cannot yet be generated.
```

These messages are displayed when the software identifies that the specified JEDEC file associated with an instance is not a supported device or does not match the specified device. Check that the specified part exists in the boundary-scan chain that you declared in your "part" command. A new "part" command will override the previously specified one. The current "part" database can be displayed by typing "part" followed by a carriage return.

```
Error 1046: The checksum calculated by reading the
programmed device values differs from the expected
result.
```

While reading back data to calculate the checksum, errors occured. Check if data security is enabled as this disables the readback functionality. If this is not the case, check for the integrity of the cable connections. If error persists you may have a bad part and Xilinx customer service should be contacted.

```
Error 1047: Xchecker re-configuration file for
boundary-scan TAP driver was not completed. Check XACT
path setting, cable connections and version of file
named 'xckjtag.sys'.
```

When the Xchecker reconfiguration file xckjtag.sys is not found or loaded correctly the above message is displayed. Check that your XACT path includes the release "data" directory and that the file "xckjtag.sys" exists in it. Also, check the integrity of the connections to the xchecker cable both at the serial port and to the target system.

```
Error 1048: The device string is not an XC95216 part
(IDCODE: string) Please verify the specification of
the order of the parts in the boundary-scan chain.
```

This message is displayed when the software identifies that the specified operation is targetting an improper device. Check that the specified part exists in the boundary-scan chain that you declared in your "**part**" command. A new "**part**" command will override the previously specified one. The current "part" database can be displayed by typing "**part**" followed by a carriage return.

Index

T

U

V

X