



XAPP139 (v1.7) February 14, 2007

Configuration and Readback of Virtex FPGAs Using JTAG Boundary-Scan

Summary

This application note demonstrates using a Boundary-Scan (JTAG) interface to configure and read back Virtex™ FPGA devices. Virtex devices have Boundary-Scan features that are compatible with IEEE Standard 1149.1. This application note is a complement to the configuration section in the Virtex data sheet and application note [XAPP138 “Virtex FPGA Series Configuration and Readback.”](#) Xilinx recommends reviewing both the data sheet and XAPP138 prior to reading this document.

Note: The information in this application note also applies to the Virtex-E FPGA family.

Introduction

The IEEE 1149.1 Test Access Port (TAP) and Boundary-Scan architecture, commonly referred to as JTAG, is a popular testing method. JTAG is an acronym for the Joint Test Action Group, the technical subcommittee initially responsible for developing the standard. This standard provides a means to ensure the integrity of individual board-level components and their interconnections. With increasingly dense multi-layer PC boards and more sophisticated surface mounting techniques, Boundary-Scan testing is becoming widely used as an important debugging standard.

Devices containing Boundary-Scan logic can send data out on I/O pins in order to test connections between devices at the board level. The circuitry can also be used to send signals internally to test the device specific behavior. These tests are commonly used to detect opens and shorts at both the board and device level.

In addition to testing, Boundary-Scan offers the flexibility for a device to have its own set of user-defined instructions. The added common vendor-specific instructions, such as configure and verify, have increased the popularity of Boundary-Scan testing and functionality.

Boundary-Scan for Virtex Devices

The Virtex family is fully compliant with the IEEE Standard 1149.1 Test Access Port and Boundary-Scan architecture. The architecture includes all mandatory elements defined in the IEEE 1149.1 Standard. These elements include the TAP, the TAP controller, the instruction register, the instruction decoder, the Boundary-Scan register, and the bypass register. The Virtex family also supports some optional instructions – the 32-bit identification register and a configuration register in full compliance with the standard. Outlined in the following sections are the details of the JTAG architecture for Virtex devices.

© 2003–2007 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. PowerPC is a trademark of IBM Inc. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Test Access Port

The Virtex TAP contains four mandatory dedicated pins as specified by the protocol ([Table 1](#)).

Table 1: Virtex TAP Controller Pins

Pin	Description
TDI	Test Data In
TDO	Test Data Out
TMS	Test Mode Select
TCK	Test Clock

Three input pins and one output pin control the IEEE 1149.1 Boundary-Scan TAP controller. In addition to the required pins, there are optional control pins such as TRST (Test Reset) and enable pins, which can be found on devices from other manufacturers. Be aware of these optional signals when interfacing Xilinx devices with devices from different vendors because these signals can need to be driven. (To determine the set of signals that must be driven to enable IEEE 1149.1 compliance, see the vendor documentation for each device on the Boundary-Scan chain.)

The TAP controller is a 16-state state machine ([Figure 1](#)). Mandatory TAP pins are as follows:

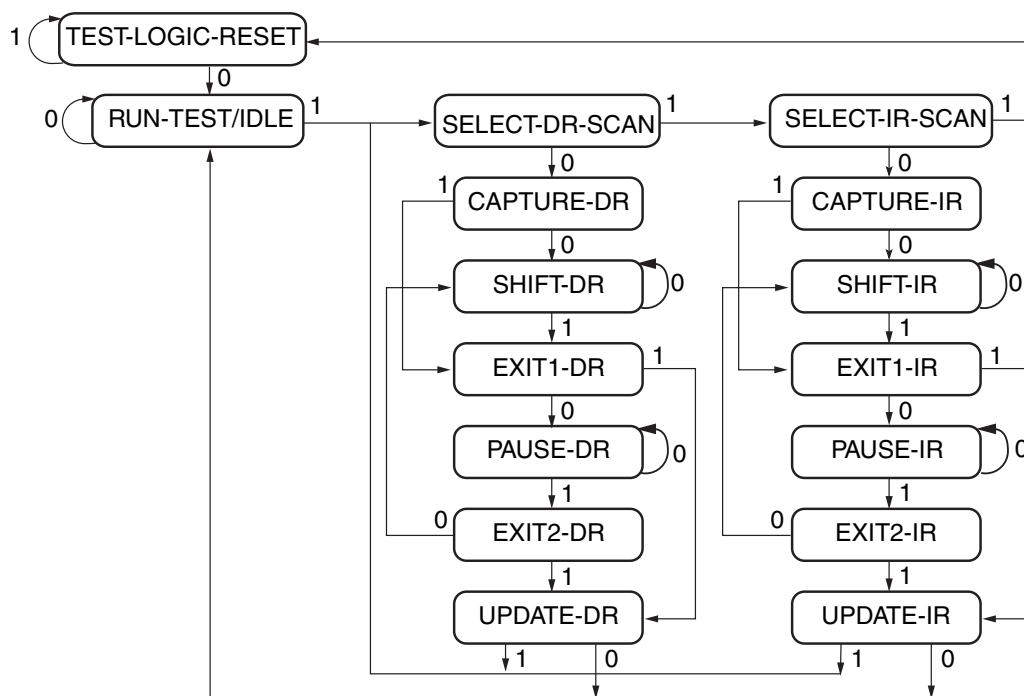
- **TMS** - The sequence of states through the TAP controller is determined by the state of the TMS pin on the rising edge of TCK. TMS has an internal resistive pull-up to provide a logic High if the pin is not driven.
- **TCK** - This pin is the JTAG test clock. It sequences the TAP controller and the JTAG registers in the Virtex devices.
- **TDI** - This pin is the serial input to all JTAG instruction and data registers. The state of the TAP controller and the current instruction held in the instruction register determine which register is fed by the TDI pin for a specific operation. TDI has an internal resistive pull-up to provide a logic High to the system if the pin is not driven. TDI is applied into the JTAG registers on the rising edge of TCK.
- **TDO** - This pin is the serial output for all JTAG instruction and data registers. The state of the TAP controller and the current instruction held in the instruction register determine which register (instruction or data) feeds TDO for a specific operation. TDO changes state on the falling edge of TCK and is active only during the shifting of instructions or data through the device. This pin is placed in a 3-state condition at all other times.

Note: As specified by the IEEE Standard, the TMS and TDI pins all have internal pull-ups. These internal pull-ups of 50-150 k Ω are active regardless of the mode selection.

When using the Boundary-Scan operations in Virtex devices, the V_{CCO} for Bank 2 must be at 3.3V for the TDO pin to operate at the required LVTTTL level.

TAP Controller

Figure 1 diagrams a 16-state finite state machine. The four TAP pins control how the data is scanned into the various registers. The state of the TMS pin at the rising edge of the TCK determines the sequence of state transitions. There are two main sequences, one for shifting data into the data register and the other for shifting an instruction into the instruction register.



Note: The value shown adjacent to each state transition in this figure represents the signal present at TMS at the time of a rising edge at TCK.

x139_01_011207

Figure 1: State Diagram for the TAP Controller

Boundary-Scan Instruction Set

To determine the operation to be invoked, a 5-bit instruction is loaded into the instruction register. Table 2 lists the available Boundary-Scan instructions for Virtex devices.

Table 2: Virtex Boundary-Scan Instructions

Boundary-Scan Command	Binary Code (4:0)	Description
EXTEST	00000	Enables Boundary-Scan EXTEST operation
SAMPLE	00001	Enables Boundary-Scan SAMPLE operation
USER1	00010	Access user-defined register 1
USER2	00011	Access user-defined register 2
CFG_OUT	00100	Access the configuration bus for readback
CFG_IN	00101	Access the configuration bus for configuration
INTEST	00111	Enables Boundary-Scan INTEST operation

Single Device Configuration

Note: Refer to [XAPP058](#) for the recommended embedded solution.

To configure a Virtex part as a single device through Boundary-Scan operations, use the steps listed in [Table 8](#), which lists and describes the TAP controller commands required to configure a Virtex device. Ensure the bitstream is generated with the JTAG clock option:

```
bitgen -g startupclk:jtagclk designName.ncd
```

Also, when programming with iMPACT software, verify that the most current version of software is used. Refer to [Figure 1](#) for the TAP controller states. These TAP controller commands are issued automatically if configuring the part with the iMPACT software.

Table 8: Single Device Configuration Sequence

TAP Controller Step Description		Set and Hold		Number of Clocks
		TDI	TMS	TCK
1	On power-up, place a "1" on the TMS and clock the TCK five times. (This ensures starting in the TLR (Test-Logic-Reset) state.)	X	1	5
2	Move into the RTI state.	X	0	1
3	Move into the SELECT-IR state.	X	1	2
4	Enter the SHIFT-IR state.	X	0	2
5	Start loading the CFG_IN instruction ⁽¹⁾	0101	0	4
6	Load the last bit of CFG_IN instruction when exiting SHIFT-IR (defined in the IEEE standard).	0	1	1
7	Enter the SELECT-DR state.	X	1	2
8	Enter the SHIFT-DR state.	X	0	2
9	Shift in the Virtex bitstream. (bit _N (MSB) is the first bit in the bitstream ⁽¹⁾)	bit ₁ ... bit _N	0	(Number of bits in bitstream) - 1
10	Shift in the last bit of the bitstream. (bit ₀ (LSB) is shifted on the transition to EXIT1-DR)	bit ₀	1	1
11	Enter UPDATE-DR state.	X	1	1
12	Enter the SELECT-IR state.	X	1	2
13	Move to the SHIFT-IR state.	X	0	2
14	Start loading the JSTART instruction ⁽¹⁾ (The JSTART instruction initializes the startup sequence.)	1100	0	4
15	Load the last bit of the JSTART instruction.	0	1	1
16	Move to the SELECT-DR state.	X	1	2
17	Move to SHIFT-DR and clock the STARTUP sequence. (by applying a minimum of 12 clock cycles to the TCK).	X	0	≥14
18	Move to the UPDATE-DR state.	X	1	2
19	Return to the RTI state. (The device is now functional).	X	0	1

Note:

1. In the TDI column, the right-most bit is shifted in first.