XAPP188 (v2.3) June 20, 2008

Configuration and Readback of Spartan-II and Spartan-IIE FPGAs Using Boundary Scan

Summary

This application note demonstrates using a Boundary-Scan (JTAG) interface to configure and read back Spartan®-II and Spartan-IIE FPGA devices. Spartan-II/IIE devices have boundary scan features that are compatible with the IEEE Standard 1149.1. This application note is a complement to the configuration section in the Spartan-II and Spartan-IIE FPGA Data Sheets and application note XAPP176: "Configuration and Readback of the Spartan-II and Spartan-IIE Families". Review of both the data sheets and XAPP176 is recommended prior to reading this document.

Introduction to JTAG

The IEEE 1149.1 Test Access Port and Boundary-Scan Architecture, commonly referred to as JTAG, is a popular testing method. JTAG is an acronym for Joint Test Action Group, the technical subcommittee initially responsible for developing the standard. This standard provides a means to assure the integrity of individual components and the interconnections between them at the board level. With increasingly dense multilayer 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 and receive data 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 device specific behavior. JTAG testing is most commonly used to detect opens and shorts at both the board and device level.

In addition to testing, IEEE standard 1149.1 Boundary Scan offers the flexibility for manufacturers to include their own set of user-defined instructions. Common vendor specific instructions, such as configure, verify, and user I.D. have increased the popularity of Boundary-Scan testing and functionality in FPGAs.

Basics and Benefits of Using Boundary Scan

Basics

- IEEE 1149.1 can be used to program and reprogram most Xilinx devices at power on or at user initiation.
- IEEE 1149.1 can be used on most Xilinx devices to read back Status, program user I.D., or configuration data.
- IEEE 1149.1 can be used to test circuit board and device connections for opens and shorts.

Benefits

- Permits the use of Automatic Test Equipment (ATE) for testing and programming.
- Supported by multiple vendors and supports mixed chains of devices.
- Provides easy access to devices internally for debugging complex issues.
- Simplifies reconfigurable system design.

Boundary Scan for Spartan-II/IIE Devices

The Spartan-II 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 Test Access Port (TAP), the TAP controller, the instruction register, the instruction decoder, the Boundary-Scan register, and the bypass register. The Spartan-II/IIE families also support some optional instructions; such as the 32-bit identification register and a configuration register. Optional instructions are in full compliance with the standard. Outlined in the following sections are the details of the JTAG architecture for Spartan-II/IIE devices.

Test Access Port

The Spartan-II/IIE TAP contains four mandatory dedicated pins as specified by the protocol (Table 1).

Table 1: **Spartan-II/IIE TAP Controller Pins**

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

In addition to these pins, the standard defines optional control pins such as $\overline{\text{TRST}}$ (Test Reset) and enable pins which are not found on Spartan-II/IIE devices. It is important to control these other signals properly when interfacing other devices with Xilinx devices.

The TAP controller is a 16-state state machine as shown in Figure 1. The four mandatory TAP pins are outlined below:

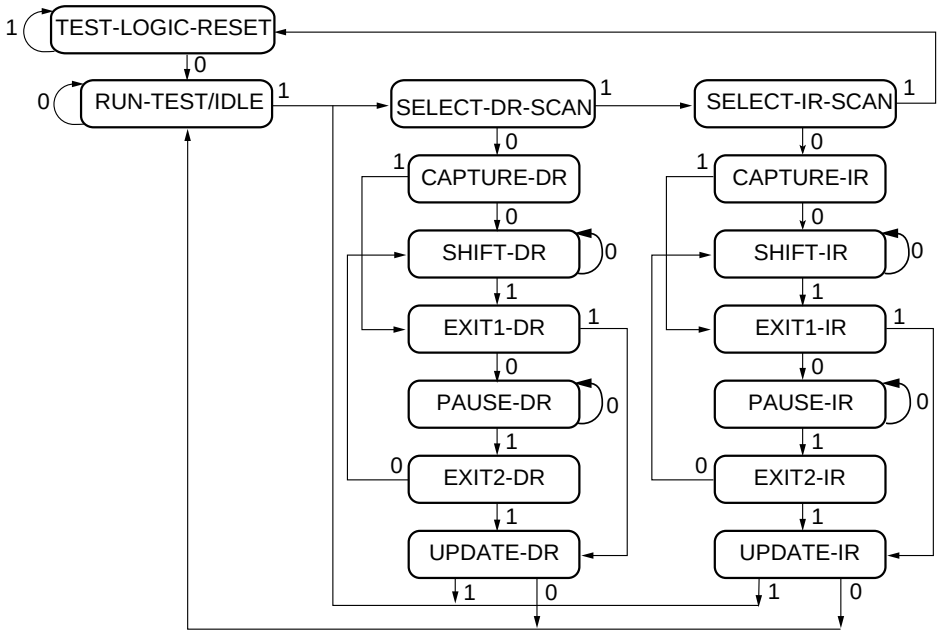
- **TMS** - The sequence of states through the TAP controller is determined by the state of the TMS pin on the rising edge of TCK. TMS has a weak internal pull-up resistor to provide a logic High if the pin is not being driven.
- **TCK** - This pin is the JTAG test clock. It sequences the TAP controller and the JTAG registers in the Spartan-II/IIE devices.
- **TDI** - This pin is the serial input to all JTAG registers. The state of the TAP controller and the current instruction held in the instruction register determine which register is the target for data coming in to the device on TDI. TDI has an internal pull-up resistor to provide a logic High to the JTAG resister if nothing is driving TDI. TDI is captured 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 only valid during the shifting of instructions or data through the device. This pin is in a high-impedance state at all other times.

  Note: As specified by the IEEE Standard, the TMS and TDI pins all have internal pull-up resistors. These internal pull-up resistors are active regardless of the mode selection.

When using the Boundary-Scan operations in Spartan-II/IIE devices, the V_{CCO} for Bank 2 must be at 3.3V for the TDO pin to operate at the required LVTTL 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 instructions into the instruction register and the other for shifting data into the data 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.

xapp188_01_122300

Figure 1: **State Diagram for the TAP Controller**

Boundary-Scan Instruction Set

To determine the operation to be invoked and which data register is to be active, a 5-bit instruction is loaded into the instruction register. Table 2 lists the available instructions for Spartan-II/IIE devices.

Table 2: **Spartan-II/IIE 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
USERCODE	01000	Enables shifting out user code
IDCODE	01001	Enables shifting out of ID code
HIGHZ	01010	Enables the bypass register while placing I/Os in a high-impedance state
JSTART	01100	Clocks the FPGA configuration start-up sequence when the start-up clock is selected as TCK
BYPASS	11111	Enables BYPASS
RESERVED	All other codes	Xilinx reserved instructions

The mandatory IEEE 1149.1 commands are supported in Spartan-II/IIE devices along with several Xilinx vendor specific commands. Spartan-II/IIE devices have a powerful command set. The EXTEST, INTEST, SAMPLE/PRELOAD, BYPASS, IDCODE, USERCODE, and HIGHZ instructions are all included. The TAP also supports two internal user-defined registers (USER1 and USER2) and configuration/readback of the device. The Spartan-II/IIE FPGA Boundary-Scan operations are independent of the mode selection. The Boundary-Scan mode in Spartan-II/IIE devices operates regardless of the other mode selections. For this reason, Boundary-Scan instructions using the Boundary-Scan register (SAMPLE/PRELOAD, INTEST, EXTEST) must not be performed during configuration. Also the $\overline{\text{INIT}}$ pin must be used to delay configuration of devices set in master mode while doing a JTAG configuration. All instructions except USER1 and USER2 are available before the Spartan-II/IIE device is configured. After configuration, all instructions are available.

JTAG Clock Option

JSTART is an instruction specific to the Spartan-II/IIE FPGA architecture and configuration flow. As described in Table 2, the JSTART instruction uses TCK to clock the startup sequence when the appropriate configuration option is selected. The instruction does not work correctly without the correct configuration option selected in the Xilinx implementation software. The command line representation for this option is:

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

For details on standard Boundary-Scan instructions, EXTEST, INTEST, and BYPASS, refer to the IEEE Standard. The user-defined registers (USER1/USER2) are described in a later section of this application note.

Boundary-Scan Architecture

Spartan-II/IIE devices have several registers associated with the IEEE standard. In addition to the standard registers, Spartan-II/IIE devices contain additional registers for simplified testing and verification (Table 3).

Table 3: **Spartan-II/IIE JTAG Registers**

Register Name	Register Length	Description
Instruction register	5 bits	Holds current instruction OPCODE and captures internal device status.
Boundary-Scan register	3 bits per I/O	Controls and observes input, output, and output enable.
Bypass register	1 bit	Device bypass.
Identification register	32 bits	Captures device ID, including Xilinx manufacturing code.
JTAG configuration register	32 bits	Allows access to the configuration bus when using the CFG_IN or CFG_OUT instructions.
USERCODE register	32 bits	Captures user-programmable code.

Boundary-Scan Register

The test primary data register is the Boundary-Scan register. The Boundary-Scan operation is independent of individual input/output block (IOB) configurations. Each IOB, bonded or unbonded, starts out as bidirectional with 3-state control. Later, it can be configured to be an input, output, or 3-state only. Therefore, three data register bits are provided per IOB (Figure 2).

When conducting a data register (DR) operation, the DR captures data in a parallel fashion during the CAPTURE-DR state. The data is then shifted out and replaced by new data during the SHIFT-DR state. For each bit of the DR, an update latch is used to hold the input data stable during the next SHIFT-DR state. The data is then latched during UPDATE-DR state when the TCK is Low.

The update latch is opened each time the TAP Controller enters the UPDATE-DR state. Care is necessary when exercising an INTEST or EXTEST to ensure the proper data has been latched before exercising the command. This is typically accomplished by using the SAMPLE/PRELOAD instruction.

Consider the presence of internal pull-ups and pull-down resistors when developing test vectors for testing opens and shorts. The IOB can be connected to an internal pull-up or pull-down resistor depending on the configuration state of the FPGA.

- For an FPGA that is not yet configured, the Spartan-II/IIE configuration mode determines whether or not to connect the IOB to an internal pull-up resistor.
- For a configured FPGA, the connectivity of an IOB to an internal pull-up or pull-down resistor depends on the user configuration of the IOB or the BitGen setting for unused pins.

Figure 2 shows the Spartan-II/IIE Boundary-Scan arcitecture.

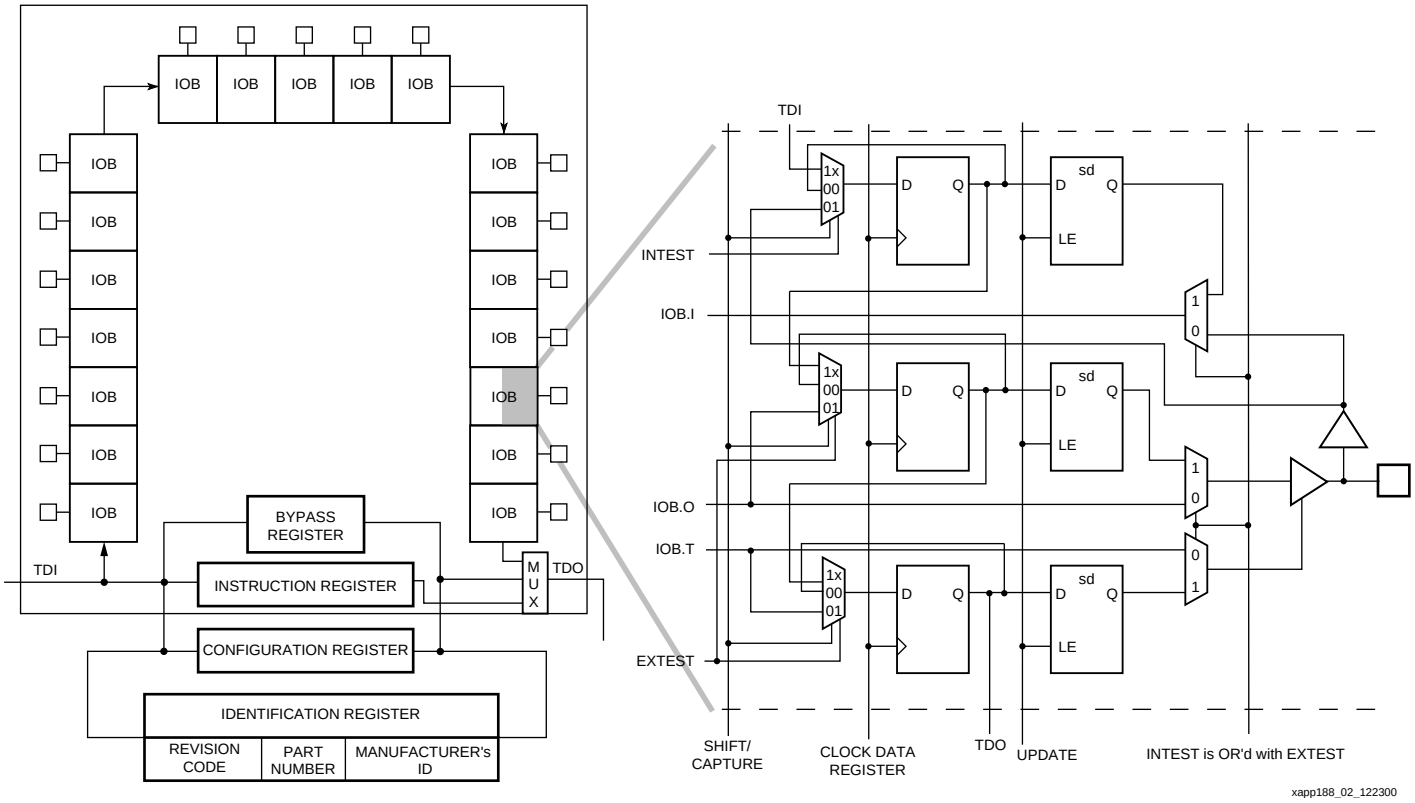


Figure 2: **Spartan-II/IIE Family Boundary-Scan Logic**

Bit Sequence

The order of registers in each regular IOB is described in this section. The input is first, followed by the output, and finally the 3-state IOB control. The 3-state IOB control is closest to the TDO. The special input-only pins on Spartan-II/IIE devices contain only the input bit to the Boundary-Scan I/O data register. The bit sequence of the device is fully defined in the “Boundary Scan Description Language Files” (BSDL) for the Spartan-II/IIE families. These files can be obtained from the Xilinx software download area (http://www.xilinx.com/support/download/index.htm).

The bit sequence is invariant of the design. It always has the same bit order and the same number of bits.

Bypass Register

The bypass register is a single flip-flop standard data register. It directly passes data serially from the TDI pin to the TDO pin during a bypass instruction. This register is initialized to “0” when the TAP controller is in the UPDATE-DR state.

Instruction Register

The instruction register is a 5-bit register that loads the OPCODE necessary for the Spartan-II/IIE FPGA Boundary-Scan instruction set. This register loads the current OPCODE and captures internal device status.

Configuration Register (Boundary Scan)

The configuration register is a 32-bit register. This register allows access to the configuration bus and readback operations.

Identification Register

The Spartan-II/IIE devices have an identification register, commonly referred to as the IDCODE register. This register is based on the IEEE Standard 1149.1 and allows easy identification of the part being tested or programmed via boundary scan. The IDCODE is the default instruction at power-up and Test-Logic-Reset state. The general format of the IDCODE in the Spartan-II/IIE family is identified in Figure 3. Specific Spartan-II/IIE IDCODEs are listed in Table 4.

Part Number			Manufacturer ID	LSB
Revision Code	**Family Code**	**Part Size Code**		
31 ... 28	27 ... 21	20 ... 12	11 ... 1	0
XXXX	0000011 (Spartan-II FPGAs) 0000101 (Spartan-IIE FPGAs)	YYYYYYYYY	0000 1001 001	1

Figure 3: **Spartan-II/IIE Identification Register**

Table 4: **IDCODEs Assigned to Spartan-II and Spartan-IIE FPGAs**

FPGA	IDCODE (Hex)	IDCODE (Binary)
XC2S15	0xX0608093	XXXX 0000 0110 0000 1000 0000 1001 0011
XC2S30	0xX060C093	XXXX 0000 0110 0000 1100 0000 1001 0011
XC2S50	0xX0610093	XXXX 0000 0110 0001 0000 0000 1001 0011
XC2S100	0xX0614093	XXXX 0000 0110 0001 0100 0000 1001 0011
XC2S150	0xX0618093	XXXX 0000 0110 0001 1000 0000 1001 0011
XC2S200	0xX061C093	XXXX 0000 0110 0001 1100 0000 1001 0011
XC2S50E	0xX0A10093	XXXX 0000 1010 0001 0000 0000 1001 0011
XC2S100E	0xX0A14093	XXXX 0000 1010 0001 0100 0000 1001 0011
XC2S150E	0xX0A18093	XXXX 0000 1010 0001 1000 0000 1001 0011
XC2S200E	0xX0A1C093	XXXX 0000 1010 0001 1100 0000 1001 0011
XC2S300E	0xX0A20093	XXXX 0000 1010 0010 0000 0000 1001 0011
XC2S400E	0xX0A28093	XXXX 0000 1010 0010 1000 0000 1001 0011
XC2S600E	0xX0A30093	XXXX 0000 1010 0011 0000 0000 1001 0011

USERCODE Register

USERCODE is supported in the Spartan-II/IIE families as well. This register allows a user to specify a design-specific identification code. The USERCODE can be programmed into the device and read back through JTAG for verification at a later time. The USERCODE is embedded into the bitstream during bitstream generation and is only valid after configuration.

USER1, USER2 Registers

The USER1 and USER2 boundary-scan instructions are valid only after configuration. The user can define data registers associated with the USER1 and USER2 instructions within the FPGA design. After the FPGA is configured, the user can access the USER1 and USER2 data registers through the TAP pins.

The BSCAN_SPARTAN2 library macro is required when creating the USER1 and USER2 data registers. This symbol is required only for driving internal scan chains (USER1 and USER2). The BSCAN_SPARTAN2 macro provides two user pins (SEL1 and SEL2) that determine the usage of USER1 or USER2 instructions, respectively.

For these instructions, two corresponding pins (TDO1 and TDO2) allow user scan data to be shifted out of TDO. In addition, there are individual clock pins (DRCK1 and DRCK2) for each user register. There is a common input pin (TDI) and shared output pins that represent the state of the TAP controller (RESET, SHIFT, and UPDATE). Unlike the earlier FPGA families where the BSCAN macro was required to dedicate the TAP pins for boundary scan, the Spartan-II/IIE FPGA TAP pins are always dedicated and do not require the BSCAN_SPARTAN2 macro for normal boundary scan instructions or operations. The BSCAN_SPARTAN2 macro is only needed for advanced JTAG features such as the USER1 and USER2 registers.

Note that USER1 and USER2 are user-defined registers. The example (Figure 4) is one of many possible implementations. For HDL, the BSCAN_SPARTAN2 macro needs to be instantiated in the design.

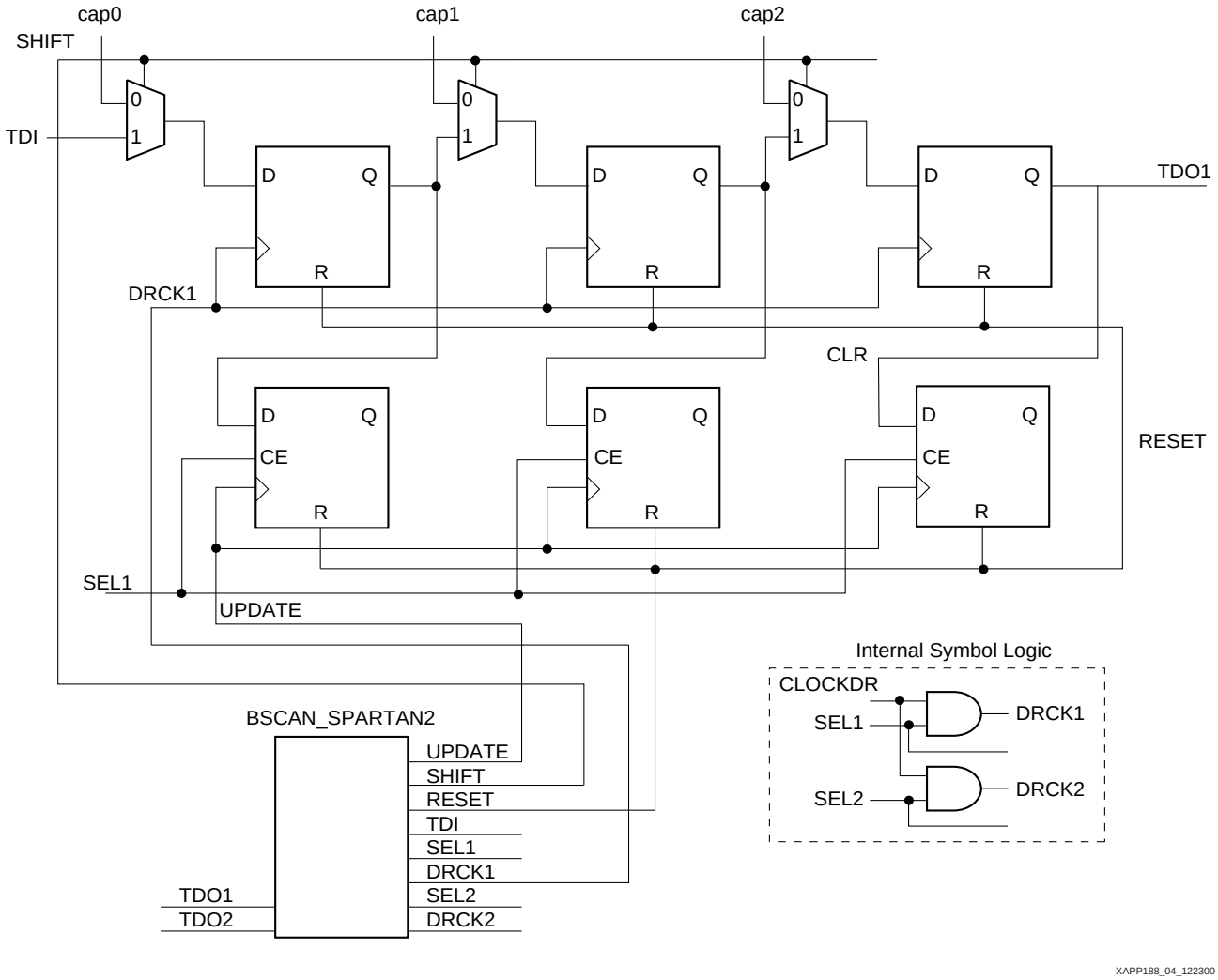


Figure 4: **BSCAN_SPARTAN2 (Example 3-bit Internal Scan Chain Usage)**

Boundary Scan in Spartan-II/IIE Devices

Refer to Application Note XAPP176: "Configuration and Readback of the Spartan-II and Spartan-IIE Families" for further information on the startup sequence, bitstream, and internal configuration registers referenced in this application note.

Configuring Through Boundary Scan

One of the most common Xilinx instructions is the configure instruction. An individual Spartan-II device is configured via boundary scan on power-up using the TAP controller. If the Spartan-II/IIE device is only configured via JTAG, it is recommended to tie the mode pins to one of the Boundary-Scan configuration mode settings; "101" (M2=1, M1=0, M0=1: contains no pull-ups on I/Os) or "001" (M2=0, M1=0, M0=1: contains pull-ups on I/Os). If using a master configuration mode and JTAG configuration mode, set the mode pins for the master mode and pull the $\overline{\text{INIT}}$ pin Low before issuing the JTAG configure instruction, hold the $\overline{\text{INIT}}$ pins Low to configure with JTAG. Table 5 shows the mode pin settings for all the configuration modes, including the Boundary-Scan modes.

Table 5: **Spartan-II/IIE Configuration Modes**

Configuration Mode	M2	M1	M0	Pull-ups
Master Serial (1)	0	0	0	NO
Slave Serial	1	1	1	NO
Boundary Scan	**1**	**0**	**1**	**NO**
Master Serial (with Pull-ups)(1)	1	0	0	YES
Slave Serial with Pull-ups)	0	1	1	YES
Serial Parallel (with Pull-ups)	0	1	0	YES
Boundary Scan (with Pull-ups)	**0**	**0**	**1**	**YES**

Notes:
1. $\overline{\text{INIT}}$ pin must be held Low to start a configure instruction when mode pins are set to master.

In addition to the mode pins, the user must be aware of or control the $\overline{\text{PROGRAM}}$, $\overline{\text{INIT}}$, and DONE configuration pins functions. The TAP Port is held in Reset if the $\overline{\text{PROGRAM}}$ pin is asserted Low. The $\overline{\text{INIT}}$ pin can be used to delay a device from configuring. The DONE pin relays the status of a device as configured or not configured (Low).

Table 6: **Spartan-II/IIE Special Configuration Pins**

Pin Name	Description
$\overline{\text{PROGRAM}}$	Initiates a configuration sequence when asserted Low. Holds JTAG TAP in Reset.
$\overline{\text{INIT}}$	When Low, indicates that the configuration memory is being cleared. This pin becomes a user I/O after configuration.
DONE	Indicates that configuration loading is complete, and that the start-up sequence is in progress. The output may be open drain.

The flow diagram for Spartan-II/IIE configuration through JTAG is shown in Figure 5. The sections that follow describe how the Spartan-II/IIE device can be configured as a single device via boundary scan or as part of a multiple-device scan chain.

A configured device can be reconfigured by toggling the TAP and entering the CFG_IN instruction after pulsing the $\overline{\text{PROGRAM}}$ pin or issuing the shut-down sequence. (Refer to the "Reconfiguring a Spartan-II/IIE Device" section.) For additional details on power-up or the start-up sequence in Spartan-II/IIE devices, refer to application note XAPP176: "Configuration and Readback of the Spartan-II and Spartan-IIE Families". In addition, application note XAPP058 "Xilinx In-System Programming Using an Embedded Microcontroller" has detailed information on using Spartan-II devices in an embedded solution.

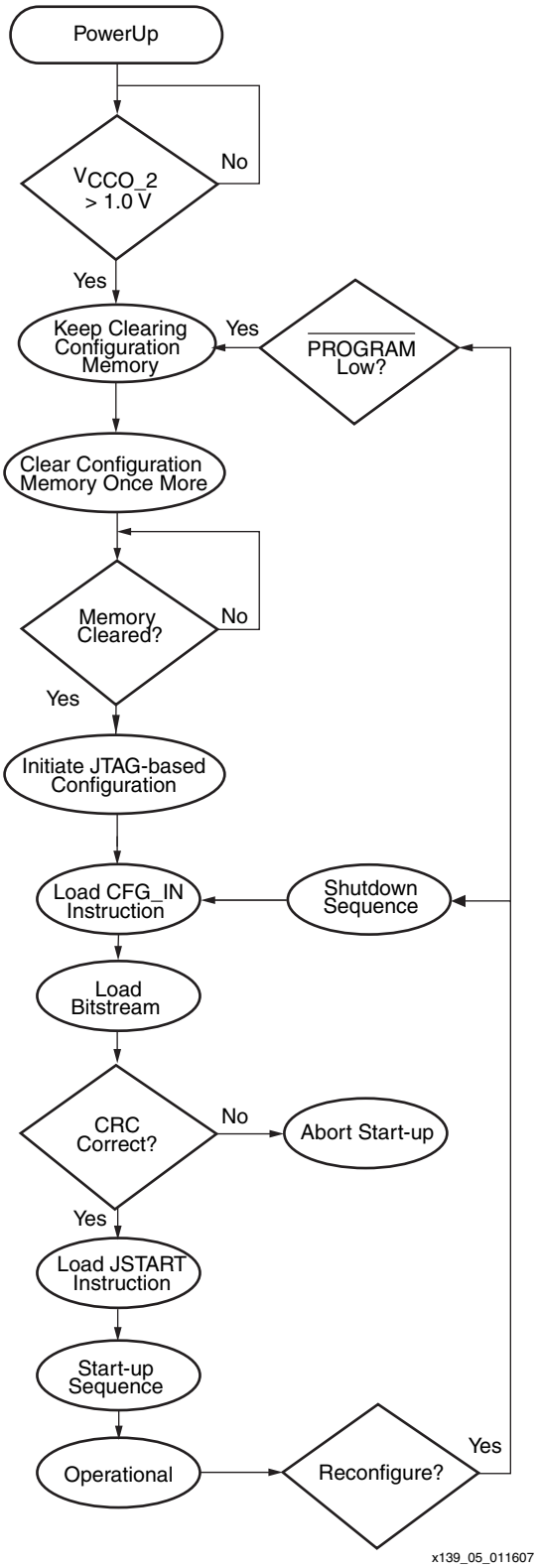


Figure 5: **Device Configuration Flow Diagram**

Single Device Configuration

To configure a Spartan-II/IIE part as a single device via Boundary-Scan operations, the following steps should be followed. Ensure the bitstream is generated with the **JTAG Clock Option**, page 4 enabled.

Also, when programming with iMPACT software or other Xilinx download tools, confirm that the most current version of software and BSDL files from support.xilinx.com are used.

Table 7 describes the TAP controller commands required to configure a Spartan-II/IIE device. Refer to Figure 1 for the TAP controller states. These TAP controller commands are issued automatically if configuring the part with Xilinx software.

Table 7: **Single Device Configuration Sequence**

TAP Controller Step Description		Set and Hold		# 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 Test-Logic-Reset state)	X	1	5
2	Move into the Run-Test-Idle 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 Spartan-II/IIE bitstream. (bit_N [MSB] is the first bit in the bitstream(1))	bit_1... bit_N	0	(# of bits in bitstream) –1
10	Shift in the last bit of the bitstream. (bit_0 [LSB] is shifted on the transition to EXIT-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. (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	≥12
18	Move to the UPDATE-DR state	X	1	2
19	Return to the Run-Test_Idle state. (The device is now functional.)	X	0	1

Notes:

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

Multiple Device Configuration

It is possible to configure multiple Spartan-II/IIE devices in a chain. The devices in the JTAG chain are configured one at a time. The multiple device configuration steps are described generally to be applied to any size chain. Ensure signal integrity on all JTAG traces. Use IBIS simulations to determine if buffering is required. Ensure the bitstream is generated with the JTAG clock option:

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

Refer to Figure 1 for the following TAP controller steps:

1. On power-up, place a “1” on the TMS and clock the TCK five times. This ensures the device will start in the Test-Logic-Reset state.
2. Load the CFG_IN instruction into the target device (and BYPASS in all other devices.)
3. For a JTAG chain, it is necessary to shift in leading “0s” before the bitstream if the value of N is not equal to “0”. Use the following equation to determine the number of leading “0s” for each bitstream. M is the targeted device position in the chain. As shown in Figure 6, the first device position in the chain is “0”. N is the number of “0s” required.

$$N = 32 - \mathrm{mod}\left(\frac{M}{32}\right)$$

$$N = 32 - M \quad \text{for} \quad (M \leq 32)$$

Example

The third device, position 2, in Figure 6 requires 30 leading “0s”.

$$N = 32 - 2 \qquad N = 30 \qquad \text{number of leading 0s}$$

The following example is for position 47:

$$N = 32 - \mathrm{mod}\left(\frac{47}{32}\right) \qquad N = 17 \text{ number of leading 0s}$$

4. Go through RUN-TEST/IDLE
5. Repeat steps 2 and 4 for each successive device.
6. Load the JSTART command into all devices.
7. Go to SHIFT-DR and clock TCK 12 times.
8. All devices are active at this point.

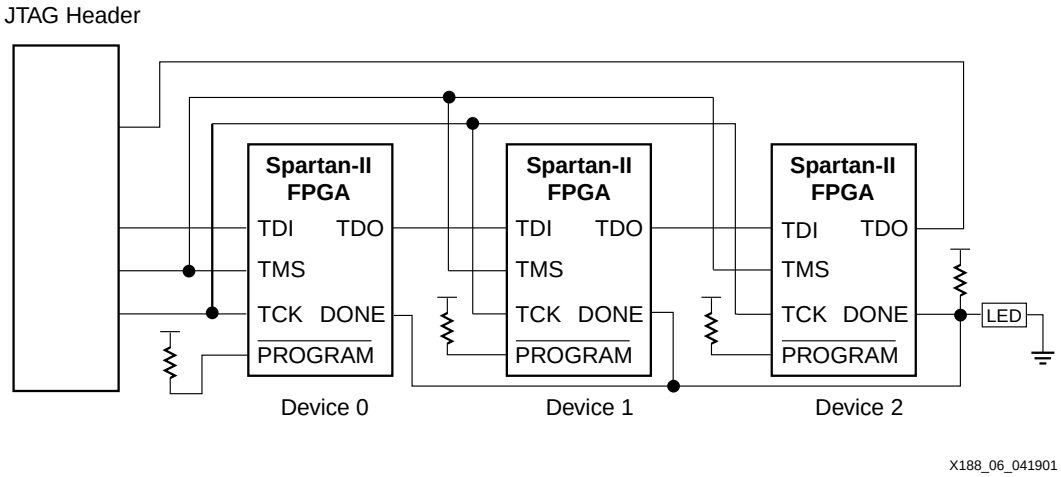


Figure 6: **Boundary-Scan Chain of Devices**

Reconfiguring Through Boundary Scan

Note: Refer to XAPP058 for the recommended embedded solution.

There are two methods to reconfigure Spartan-II/IIE devices without possible internal contention. The first method is to pulse the PROGRAM pin. The alternate method is to perform a shutdown sequence, placing the device in a safe state. The following shutdown sequence includes using internal registers. (For details on internal registers, refer to Application Note: XAPP176: "Configuration and Readback of the Spartan-II and Spartan-IIE Families".)

1. Load the CFG_IN instruction into the JTAG instruction register. Next, go to the SHIFT-DR state. If reconfiguring a chain of multiple devices, refer to step 6 (Multiple Device Configuration) on page 12.
2. In the SHIFT-DR state, shift in the following sequences in steps 2 through 4. (This writes the COR (Configuration Option Register) with the SHUTDOWN bit = 1. It also indicates that the startup sequencer should perform a shutdown sequence.) The most significant bit (MSB) is the left bit and it is shifted in first.

   ```
   0011 0000 0000 0001 0010 0000 0000 0001
   -> Header: Write to COR

   0000 0000 1010 0000 1011 1111 0010 1101
   -> COR data sets SHUTDOWN = 1
   ```

3. Write the START command to the CMD (Command) register by shifting in the following data:

   ```
   0011 0000 0000 0000 1000 0000 0000 0001
   -> Header: Write to CMD

   0000 0000 0000 0000 0000 0000 0000 0101
   -> START command
   ```

4. Write the precalculated CRC value to the CRC (Cyclic Redundancy Check) register, or write the RCRC (Reset CRC Register) command to the CMD register as shown:

   ```
   0011 0000 0000 0000 1000 0000 0000 0001
   -> Header: Write to CMD

   0000 0000 0000 0000 0000 0000 0000 0111
   -> RCRC command

   0000 0000 0000 0000 0000 0000 0000 0000
   -> flush pipe
   ```

5. Now proceed to the SHIFT-IR and load the JTAG JSTART command into the instruction register.
6. Go to the SHIFT-DR and clock TCK 12 times to clock the shutdown sequence.
7. Proceed to the SHIFT-IR state and load the CFG_IN instruction again.
8. In the SHIFT-DR state, shift in the sequences in steps 8 and 9. This writes the AGHIGH command to the CMD register to assert the GHIGH_B signal. This prevents contention while writing configuration data. If reconfiguring a chain, refer to step 6 on page 12.

   ```
   0011 0000 0000 0000 1000 0000 0000 0001
   -> Header: Write to CMD

   0000 0000 0000 0000 0000 0000 0000 1000
   -> AGHIGH command asserts GHIGH
   ```

9. Write the COR with SHUTDOWN = 0 and go to RUN-TEST/IDLE by shifting in the following sequence:

   ```
   0011 0000 0000 0001 0010 0000 0000 0001
   -> Header: Write to COR
   ```

```
0000 0000 1010 0000 0011 1111 0010 1101
-> COR data sets SHUTDOWN = 0

0000 0000 0000 0000 0000 0000 0000 0000

-> flush pipe
```

Debugging Configuration

To verify successful configuration, there are several options. Some of the most helpful verification steps include using the TAP controller pins and the readback command. Using the Spartan-II/IIE FPGA TAP controller and status pins is discussed first.

When using the TAP controller pins, TDO is only driven in the SHIFT-DR and SHIFT-IR state. If the output of the TDO can be changed via an external pull-up, the TAP controller is not in SHIFT-IR or SHIFT-DR. If the TAP can be controlled precisely, use this to test the application.

In JTAG configuration, the status pin (DONE) functions the same as in the other configuration modes. The DONE pin can be monitored to determine if a bitstream has been completely loaded into the device. If DONE is Low, the entire bitstream has not been sent or the start-up sequence is not finished. If DONE is High, the entire bitstream has been received correctly. When the FPGA detects a bitstream CRC error, DONE remains Low and INIT is driven Low.

If the DONE pin is not asserted High, there are several possible reasons.

1. The bitstream option, `-g startupclk:jtagclk` described in **Software Support and Data Files**, page 15, may not have been used.
2. The JSTART instruction was not issued.
3. There was an error in the bitstream.

In addition to the external pin monitoring, an internal test can be conducted. The second method includes the following steps to capture the contents of the internal device status register:

1. Move the TAP controller to the Test-Logic-Reset state.
2. Go to the SHIFT-IR state and load in the CFG_IN instruction.
3. Go to the SHIFT-DR state and shift in the following 64-bit pattern with the MSB (left most bit), shifted in first.

   ```
   0010 1000 0000 0000 1110 0000 0000 0001

   -> Header: Read Status Register

   0000 0000 0000 0000 0000 0000 0000 0000

   -> flush pipe
   ```

4. After shifting in this pattern, load the CFG_OUT instruction in the SHIFT-IR state.
5. Move to the SHIFT-DR and clock TCK 32 times while reading TDO. The data seen on TDO is the contents of the status register (discard the first bit). The last bit out is a “1” if a CRC error occurred. If successful, it should be:

   ```
   0000 0000 0000 0000 0111 1111 0101 1110
                                      CRC Bit
   ```

The device status register also gives the status of the DONE and INIT signals. For information on the status register, refer to Figure 7 and the Application Note: XAPP151, “Virtex® Series Configuration Architecture User Guide”. Note that the status register architecture in Spartan-II/IIE devices are the same as Virtex devices.

Refer to Application Note: XAPP104: "A Quick JTAG ISP Checklist" for general techniques on how to reduce noise and signal degradation in JTAG chains.

																DONE	INIT	MODE			GHIGH_B	GSR_B	GWE_B	GTS_CFG	IN_ERROR	LOCK				CRC_ERROR
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Notes:

1. An "X" in a bit field indicates that the value is variable and must be set.
2. Heavy vertical lines are used to separate fields. Light vertical lines separate nibbles in the word.

Figure 7: **Status Register Fields**

Readback Instructions

Readback is available through both the Boundary-Scan and Slave Parallel interfaces. It is the process of verifying that the current configuration data in the device is correct by reading out the data in the internal configuration memory. Note that readback is supported in the iMPACT software, but only in the form of the configuration verify operation. For readback/verify, ensure the bitstream is generated with the **Readback Option**.

Readback Verify

Readback verification is not unique to JTAG devices. This section discusses Boundary-Scan specific steps to perform a readback/verify operation. For detailed information on readback/verify not specific to JTAG, refer to Application Note: XAPP176: "Configuration and Readback of the Spartan-II and Spartan-IIE Families".

1. Load in the CFG_IN instruction into the JTAG IR and then go to the SHIFT-DR.
2. Shift in a packet to write the starting frame address into the FAR (Frame Address Register). For a full-chip readback, this is frame zero of the CLB column zero.
3. Shift in a packet to write the RCFG (Read Configuration Data) command into the CMD register.
4. Shift in a packet header requesting a read of the Frame Data Output Register (FDRO). The word count should reflect the number of frames you wish to read.
5. Shift in an extra 32 bits. These are to flush the pipeline through the packet processor. Then go back to RUN-TEST/IDLE.
6. Load the CFG_OUT instruction into the JTAG IR and then go to the SHIFT-DR.
7. Clock TCK and read TDO. There is one word plus one frame of garbage before the readback data appears.
8. To read the block RAM data, repeat steps one through seven by substituting the appropriate block RAM address for the starting CLB address. It is recommended that the system be halted prior to block RAM readback.

Software Support and Data Files

For the current version of iMPACT software that supports the Spartan-II/IIE devices go to the Xilinx website: http://www.xilinx.com/support/download/index.htm. The Xilinx tool set includes the iMPACT software to program and get the Spartan-II/IIE IDCODE. For test vectors EXTEST or INTEST, or to utilize other JTAG features present in the device see http://www.xilinx.com/products/design_resources/config_sol/resource/isp_ate.htm for third party Boundary-Scan software tools.

Note: To perform any configuration operations through JTAG, the file must be generated with the start-up clock set for the **JTAG Clock Option**, page 4.

Quick Reference Steps

Below is a summary for Configuration and Readback using Boundary Scan.

1. Include BSCAN in design if user registers are required.
2. Generate bitstream
   a. Use TCK for start-up
   b. Integrate into JTAG data (modify as necessary for daisy chain, etc.)
3. Connect JTAG pins on board
4. Set mode pins to JTAG or disable master mode by holding $\overline{\text{INIT}}$ Low.
5. On power-up, load configuration instruction
6. Enter shift state
7. Shift bitstream
8. Load startup instruction and clock 12 or more times
9. FPGA is operational (DONE pin should be High)

Bibliography

The following publications contain information about the IEEE Standard 1149.1 which should be consulted for general Boundary-Scan information beyond the scope of this application note.

Colin M. Maunder and Rodham E. Tulloss. *The Test Access Port and Boundary-Scan Architecture (IEEE)*. IEEE Computer Society Press, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1264.

Ken Parker. *The Boundary-Scan Handbook*. Kluwer Academic Publications, (781) 871-6600.

IEEE Standards, standards.ieee.org

Texas Instruments, IEEE Std 1149.1 (JTAG) Testability Primer

JTAG Technologies, www.jtag.com

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/23/00	1.0	Initial Xilinx release.
04/19/01	2.0	Updated.
03/27/02	2.1	Updated.
06/24/05	2.2	Added Spartan-IIE IDCODE values.
06/20/08	2.3	Extended description of Boundary Scan Register and USER1/USER2 Registers. Corrected hex equivalent for Spartan-IIE IDCODEs in Table 4. Updated Figure 5. Updated links.