XILINX®

Boundary-scan in XC4000, Spartan™ and XC5200 Series Devices

XAPP017 (v3.0) November 16, 1999

Application Note

Summary

XC4000, Spartan and XC5200 series FPGA devices contain boundary-scan facilities that are compatible with IEEE Standard 1149.1. This Application Note describes those facilities in detail, and explains how Boundary-scan is incorporated into an FPGA design.

Introduction

In production, boards must be tested to assure the integrity of the components and the interconnections. However, as integrated circuits have become more complex and multi-layer PC-boards have become more dense, it has become increasingly difficult to test assembled boards.

Originally, manufacturers used functional tests, applying stimuli to the input connectors of the board, and observing the results at the output. Later, "bed-of-nails" testing became popular, where a customized fixture presses sharp, nail-like stimulus and test-probes into the exposed traces on the board. These probes were used to force signals onto the traces and observe the response.

However, increasingly dense multi-layer PC boards with ICs (Integrated Circuits) surface-mounted on both sides have stretched the capability of bed-of-nail testing to its limit, and the industry is forced to look for a better solution. Boundary-scan techniques provide that solution.

The inclusion of boundary-scan registers in ICs greatly improves the testability of boards. Boundary-scan provides a mechanism for testing component I/Os (Inputs and Outputs) and inter-connections, while requiring as few as four additional pins and a minimum of additional logic in each IC. Component testing may also be supported in ICs with self-test capability.

Devices containing boundary-scan have the capability of driving or observing the logic levels on I/O pins by utilizing the TAP (Test Access Port). The four pins that make up the TAP are TDI (Test Data Input), TDO (Test Data Output), TMS (Test Mode Select), and TCK (Test Clock). To test the external interconnect, devices drive values at their outputs and observe input values received from other devices. An external test controller compares the received data with expected results. Data to be driven onto outputs is distributed through a chain of shift registers. The resulting data present on the I/O pins can then be captured through the output of the same shift register path.

Data is passed serially from one device to the next, thus forming a boundary-scan path or chain from TDI (Test Data Input) that originates at the test controller and returns there through TDO (Test Data Output). Any device can be temporarily removed from the boundary-scan path by bypassing its internal shift registers, and passing the serial data directly to the next device.

XC4000, Spartan and XC5200 FPGA devices contain boundary-scan registers that are compatible with the IEEE Standard 1149.1 that was derived from a proposal by the Joint Test Action Group (JTAG). External (I/O and interconnect) testing is supported; there is also limited support for internal self-test.

Overview of XC4000, Spartan and XC5200 Boundary-Scan Features

XC4000, Spartan and XC5200 devices support the mandatory boundary-scan instructions specified in the IEEE Standard 1149.1. A Test Access Port (TAP) and instruction and data registers are provided to implement the EXTEST, SAMPLE/PRELOAD and BYPASS instructions. The TAP also supports device CONFIGURATION and READBACK as well as two USERCODE instructions for internal logic. The XC4000XLA, XC4000XV and SpartanXL series parts also have an IDCODE instruction.

Note: Prior to configuration, care must be taken not to inadvertently toggle the TAP pins, as this can activate the boundary-scan circuitry resulting in unwanted EXTEST or CONFIGURE operations.

The basic boundary-scan instructions are always available prior to configuration. After configuration, the basic instructions and the USER1/USER2 instructions are available only if boundary-scan has been specified in the schematic design or HDL code (i.e., the BSCAN macro). While SAMPLE/PRELOAD and BYPASS are available during configuration, it is recommended that boundary-scan operations not be performed during this transitory period.

Boundary-scan IOB Test Data Register operation is independent of user IOB configuration and package type. All IOBs, including those unbonded, are treated as independently controlled bidirectional boundary-scan pins. Retaining the bidirectional test capability even after configuration affords tremendous flexibility for interconnect testing.

The following description assumes that the reader is familiar with boundary-scan testing and the IEEE Standard. Only issues specific to the XC4000, Spartan and XC5200 implementation are discussed in detail. For general information on boundary-scan, please refer to the bibliography.

Deviations from the IEEE Standard

The XC4000, Spartan and XC5200 boundary-scan implementation deviates from the IEEE standard in that three dedicated pins (CCLK, PROGRAM and DONE) are not included in the boundary-scan chain.

It should also be noted that the IOB Test Data Register contains three Xilinx test bits (BSCANT.UPD, TDO.O and TDO.T) and some bits of the register may correspond to unbonded or unused pins.

Additionally, the EXTEST instruction incorporates INTEST-like functionality that is not specified in the standard. During EXTEST, the global clock buffer inputs in the corners are not disabled. This is not recommended by the standard.

The TAP pins (TMS, TCK, TDI and TDO) have bits in the IOB test data register scan chain, but connections to the TAP controller are made directly at the pads. Consequently, the operation of the TAP controller cannot be affected by boundary-scan IOB test data register content.

When the TAP is in the Shift-DR state, the contents of all data registers are shifted regardless of the instruction (including BYPASS); when in the middle of shifting data from the data register, all data must be shifted out before switching to the instruction or bypass register. This is corrected in the XC4000XLA, XC4000XV and SpartanXL families.

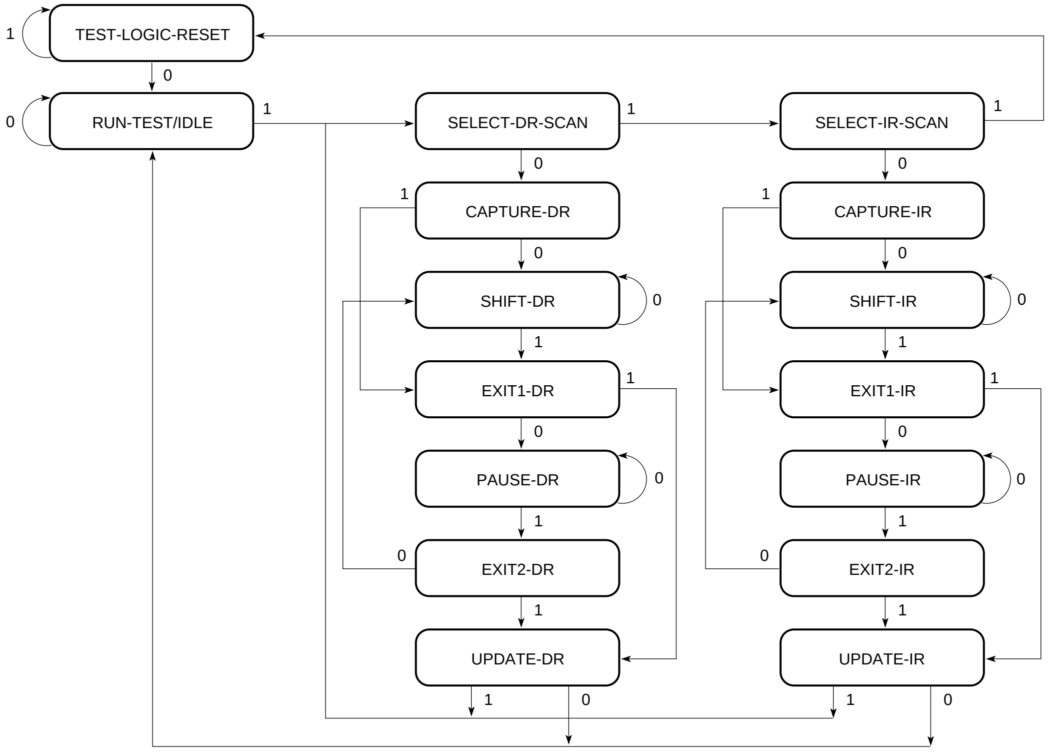
Boundary-Scan Hardware

Test Access Port

The boundary-scan logic is accessed through the Test Access Port (TAP). The TAP is comprised of four semi-dedicated pins: Test Mode Select (TMS), Test Clock (TCK), Test Data Input (TDI) and Test Data Output (TDO), as defined in the IEEE specification.

TAP Controller

The TAP Controller is a 16-state machine that controls the operation of the boundary-scan circuitry in response to the control signal TMS. This state machine, clocked by TCK, implements the state diagram specified by the IEEE standard (Figure 1). Upon power-up, the TAP controller defaults to the Test-Logic-Reset state. After configuration, the controller becomes disabled, unless its use is explicitly specified in the user design. Toggling $\overline{\text{PROGRAM}}$ resets the latched decodes for the EXTEST, CONFIGURE, and READBACK instructions.



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.

X2680

Figure 1: State Diagram for the TAP Controller N

Loading a 3-bit instruction into the Instruction Register (IR) determines the subsequent operation of the boundary-scan logic Table 1. The instruction selects the source of the TDO pin, and selects the source of IOB input and output data (IOB Test Data Register or input pin/user logic).

Table 1: Boundary-scan Instructions

Instruction			Test Selected	TDO Source	I/O Data Source
I_2	I_1	I_0			
0	0	0	EXTEST	DR	DR
0	0	1	SAMPLE/ PRELOAD	DR	Pin/Logic
0	1	0	USER 1	BSCAN.TDO1	User Logic
0	1	1	USER 2	BSCAN.TDO2	User Logic
1	0	0	READBACK	Readback Data	Pin/Logic

Table 1: Boundary-scan Instructions (Continued)

Instruction			Test Selected	TDO Source	I/O Data Source
I_2	I_1	I_0			
1	0	1	CONFIGURE	DOUT	Disabled
1	1	0	IDCODE XC4000XLA/ XC4000XV/SpartanXL only	Idcode Register	User Logic
1	1	1	BYPASS	Bypass Register	—

Note: I_0 is closest to TDO

The instruction register is used not only to hold the current instruction, but also as a status bit. If the TAP is in the Capture-IR state and TCK goes High, the instruction register captures the current status bit. For the 3-bit IR, I_0 is "1" by default, I_1 is "0" by default and I_2 is the status bit. Note that I_0 is shifted out of TDO first, then I_1, and then I_2.

The IOB Test Data Register

The IOB Test Data Register (IOBDR) is a serial shift register implemented inside the IOBs of the FPGA device. Since each IOB can be configured as an independently controlled bidirectional pin, three data registers are provided per IOB: for input data, output data and 3-state control. In practice, many of these bits go unused, but they remain in the scan chain (see Figure 2).

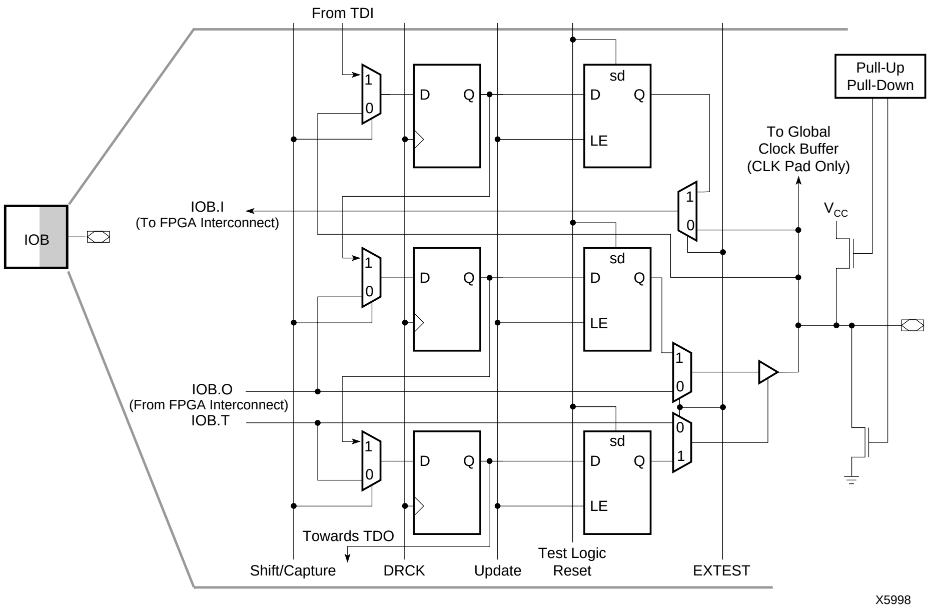


Figure 2: Boundary-scan Logic in a Typical IOB

An update latch accompanies each bit of the DR, and is used to hold input test data stable during the next DR shift. The update latch is opened during the Update-DR state of the TAP Controller when TCK is Low.

In a typical DR operation, the DR captures (parallel loads) data during the Capture-DR state (on the rising edge of TCK). This data is then shifted out and replaced with new test data from TDI during the Shift-DR state. Subsequently, the update latch opens, and the new test data becomes available for injection into the logic or the interconnect. The injection of data occurs only if the current instruction is EXTEST.

Note: The update latch is opened whenever the TAP Controller is in the Update-DR state, regardless of the instruction. Care must be taken to ensure that appropriate data is contained in the update latch prior to initiating an EXTEST. Any DR instruction, including BYPASS, that is executed after the test data is loaded, but before the EXTEST commences, changes the test data. This is corrected in the XC4000XLA, XC4000XV and SpartanXL families.

The IEEE Standard does not require the ability to inject data into the on-chip system logic and observe the results during EXTEST. However, this capability is supported to help compensate for the lack of INTEST. Test Data Register content may be initialized to specific levels in the Shift-DR state of a SAMPLE/PRELOAD or EXTEST instruction and the resulting logic outputs applied during a subsequent EXTEST. It must be recognized, however, that all IOBDR bits are captured during an EXTEST and, therefore, may change upon a subsequent update.

Pull-up and pull-down resistors remain active during boundary-scan. Before and during configuration, all pins are pulled up. After configuration, the IOB can be configured with a pull-up resistor, a pull-down resistor or neither.

Note: Internal pull-up/pull-down resistors must be taken into account when designing test vectors to detect open circuit PCB traces.

The primary and secondary global clock inputs (PGCK[1:4] and SGCK[1:4] in XC4000 and Spartan, GCK[1:8] in XC4000X and SpartanXL, and GCK[1:4] in XC5200) are taken directly from the pins, and cannot be overwritten with boundary-scan data. However, if necessary, it is possible to drive the clock input from boundary-scan. The external clock source is 3-stated, and the clock net is driven with boundary-scan data through the output driver in the clock-pad IOB. If the clock-pad IOBs are used for non-clock signals, the data may be overwritten normally.

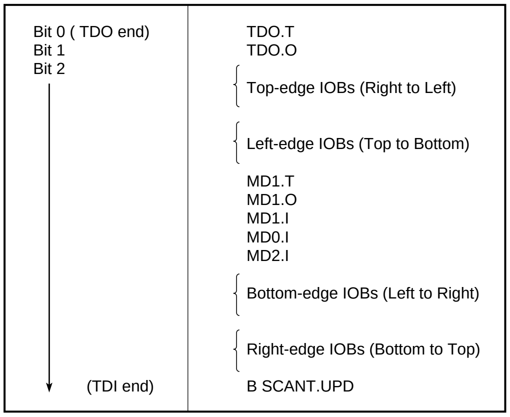
Figure 3 shows the data-register cell for a TAP pin. An OR-gate permanently disables the output buffer if boundary-scan operation is selected. Consequently, it is impossible for the outputs in IOBs used by TAP inputs to conflict with TAP operation. TAP data is taken directly from the pin, and cannot be overwritten by injected boundary-scan data.

Bit Sequence

Each non-TAP IOB corresponds to three bits in the IOBDR. The 3-state control is first (closest to TDO), the output is next, and the input is last. Other signals correspond to individual register bits. IOB locations assume that the die is viewed from the top, as in the device-level editors XDE or FPGA Editor. In the XC4000, the input-only M0 and M2 mode pins contribute only the In bit to the boundary-scan I/O register.

Table 2 lists, in data-stream order, the boundary-scan cells that make up the IOBDR for the XC4000 Series. The cell closest to TDO corresponds to the first bit of the data-stream, and is at the top of the table. Tables in the data sheets show the boundary-scan order and number of boundary-scan cells in each scan chain for all the XC4000, Spartan and XC5200 packages. Boundary-scan Description Language files are available for each part and package from the Xilinx Support web site at: http://support.xilinx.com.

Table 2: XC4000 Boundary-scan Order



Bit 0 (TDO end)	TDO.T
Bit 1	TDO.O
Bit 2	Top-edge IOBs (Right to Left)
	Left-edge IOBs (Top to Bottom)
	MD1.T MD1.O MD1.I MD0.I MD2.I
	Bottom-edge IOBs (Left to Right)
	Right-edge IOBs (Bottom to Top)
(TDI end)	B SCANT.UPD

X2674

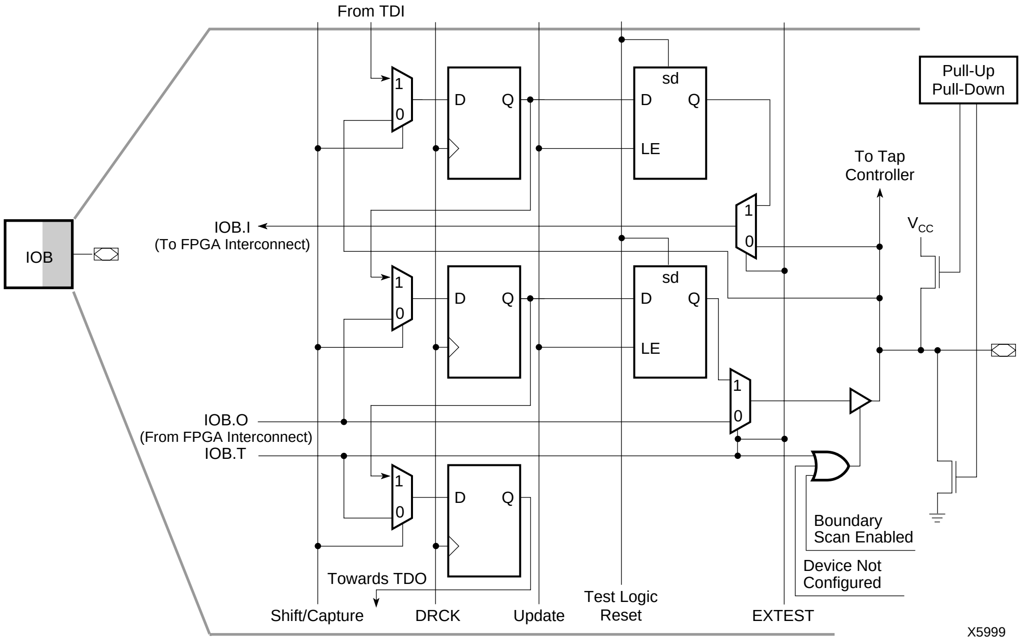


Figure 3: Boundary-scan Logic of an IOB Containing a TAP Input (TMS, TCK, and TDI Only)

Note: All IOBs remain in the test data register, independent of whether they are actually used, or even bonded. Three-bits, BSCANT.UPD, TDO.O and TDO.T, are included for Xilinx test purposes, and may be ignored by other users. In the XC4000XLA, XC4000XV and SpartanXL families TDO.O and TDO.T can provide FINISHED or ERROR status to verify configuration status. CCLK, PROGRAM and DONE are not included in the boundary-scan.

The Bypass Register

This is a 1-bit shift register that passes the serial data directly to TDO when BYPASS is the current instruction. It is initialized to 0 in the Update-DR state.

User Registers

The XC4000, Spartan, and XC5200 boundary-scan instruction set includes two USERCODE instructions, USER1 and USER2. Connections are provided to the TAP and TAP controller that, together with direct connections to the TAP pins, permit the user to include boundary-scan self-test features in the design.

The XC4000 and Spartan boundary-scan symbols have six connections for user registers: SEL1, SEL2, TDO1, TDO2, DRCK and IDLE. TDI is available directly from the input of the IOB that provides the TDI pin. The XC5200 boundary-scan symbol has three additional connections which make the creation of a user register easier: RESET, UPDATE, and SHIFT.

Note: The TDI, TMS and TCK signals supplied from IOB inputs to user test logic are overwritten by boundary-scan test data during EXTEST. During USER1 and USER2 tests, it is not altered.

SEL1, SEL2. SEL1 and SEL2 enable user logic. They are asserted (High) when the instruction register contains instructions USER1 and USER2, respectively.

TDO1, TDO2. TDO1 and TDO2 are inputs to the TDO output multiplexer, permitting user access to the CLB Logic serial boundary-scan output. They are selected when executing the instructions USER1 and USER2, respectively. Input to user data registers can be derived directly from the TDI pin, thus completing the boundary-scan chain.

There is a one flip-flop delay between TDO1/TDO2 and the TDO output. This flip-flop is clocked on the falling edge of TCK.

DRCK. Data register clock (DRCK) is a gated and uninverted version of TCK and is active only in the Capture-DR and Shift-DR states of the TAP Controller. It is provided to clock user CLB logic test-data registers. TDI data should be sampled with the rising edge of DRCK (rising edge of TCK). The TDO output flip-flop accepts data on the falling edge of DRCK (falling edge of TCK). When not active, DRCK is High.

IDLE. IDLE is a second gated version of TCK. It is active during the RUN-TEST/IDLE state of the TAP controller, and may be used to clock user test logic a set number of times, determined through TMS by the central test controller.

RESET. This signal is only available on the XC5200 boundary-scan symbol. Whenever the TAP is in the TEST-LOGIC-RESET state, the RESET pin is High, in all other cases the RESET pin is Low.

UPDATE. This signal is only available in the XC5200 boundary-scan symbol. Whenever the USER1 or USER2 instructions are used, UPDATE is an inverted version of TCK. In all other cases, UPDATE is Low.

SHIFT. This signal is only available in the XC5200 boundary-scan symbol. When the USER1 or USER2 instructions are used, SHIFT is High, in all other cases SHIFT is Low.

Using Boundary-scan

Full access to the built in boundary-scan logic is always available after device power up and before the start of configuration. During configuration, reduced boundary-scan capabilities are available: SAMPLE/PRELOAD and BYPASS. After configuration, the built-in boundary-scan logic and user test logic are available only if the boundary-scan symbol (BSCAN) is specified in the design.

Figure 6 is a flow chart of the XC4000 FPGA start up sequence that shows when the boundary-scan instructions are available. Since the PROGRAM pin resets the TAP controller, boundary-scan operations cannot commence until PROGRAM has been taken High.

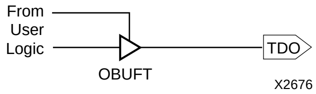


Figure 4: Typical Non-boundary-scan TDO Connection

Full boundary-scan capabilities are available after device power up until INIT is High. Without external intervention, INIT automatically goes high approximately one ms after the PROGRAM pin goes High. If more time is required for boundary-scan testing, INIT may be held Low externally until boundary-scan testing is complete. Once INIT goes High, all clocks on TCK are counted as configuration clocks for data and length count. See "Configure" on page 11 for specific details on device configuration.

Accessing boundary-scan after the device has been configured requires that the BSCAN symbol be instantiated/inserted into the design with the correct design entry syntax (see Figure 7).

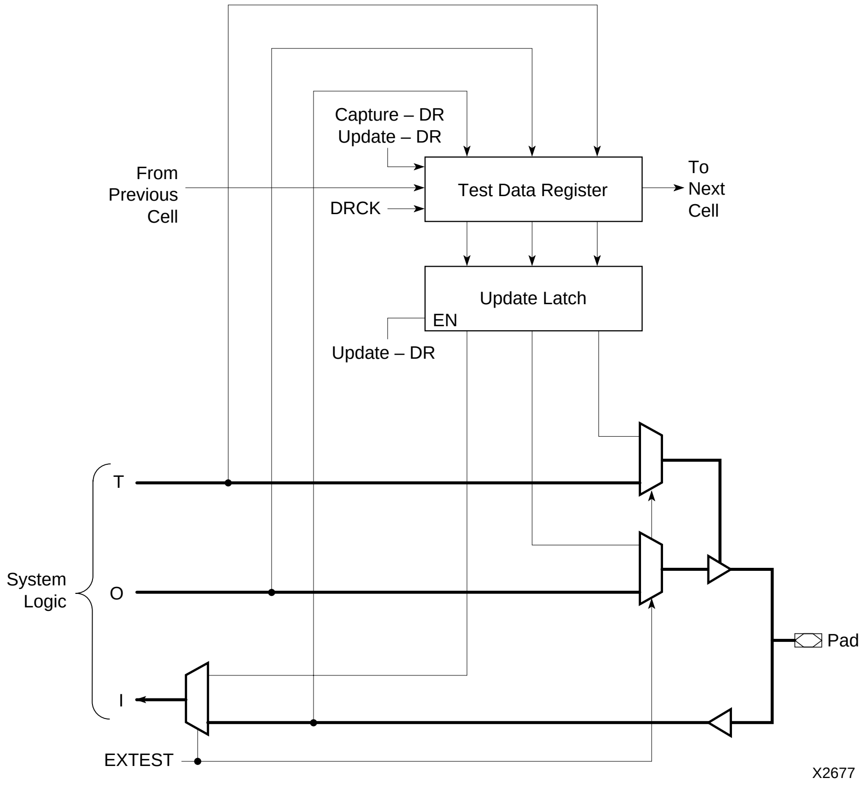


Figure 5: EXTEST Data Flow

If the BSCAN symbol is not enabled after configuration, boundary-scan is not selected, and the IOBs used by the TAP input pins are freely available as general purpose IOBs. The TDO output pin may be used as a logic output by explicitly connecting the TDO pad primitive to an OBUF or OBUFT as required (see Figure 4).

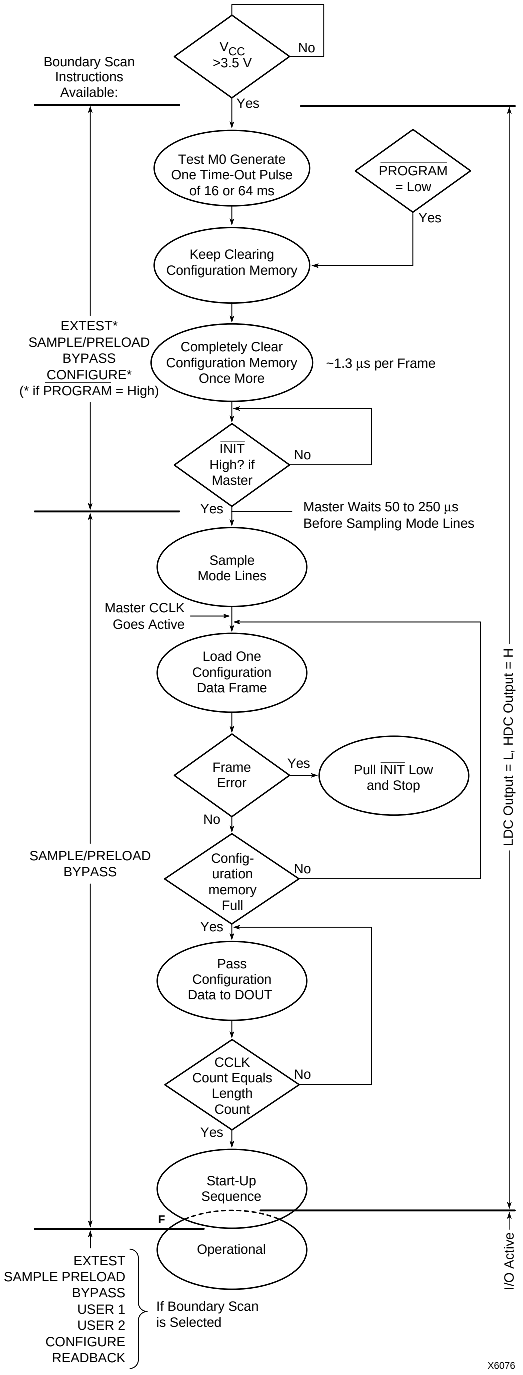

VCC >3.5 V
No
Yes
Boundary Scan Instructions Available:
Test M0 Generate One Time-Out Pulse of 16 or 64 ms
PROGRAM = Low
Yes
Keep Clearing Configuration Memory
EXTEST*
SAMPLE/PRELOAD
BYPASS
CONFIGURE*
(* if PROGRAM = High)
Completely Clear Configuration Memory Once More
~1.3 µs per Frame
INIT High? if Master
No
Yes
Master Waits 50 to 250 µs Before Sampling Mode Lines
Sample Mode Lines
Master CCLK Goes Active
Load One Configuration Data Frame
Frame Error
Yes
Pull INIT Low and Stop
No
SAMPLE/PRELOAD
BYPASS
Config-uration memory Full
No
Yes
Pass Configuration Data to DOUT
CCLK Count Equals Length Count
No
Yes
Start-Up Sequence
F
Operational
LDC Output = L, HDC Output = H
I/O Active
EXTEST
SAMPLE PRELOAD
BYPASS
USER 1
USER 2
CONFIGURE
READBACK
If Boundary Scan is Selected
X6076

Figure 6: XC4000 Startup Sequence

Boundary-scan Instructions

The XC4000/Spartan/XC5200 boundary-scan supports three IEEE-defined instructions (EXTEST, SAMPLE/PRELOAD and BYPASS), two user-definable instructions (USER1 and USER2), and two FPGA-specific instructions (CONFIGURE and READBACK). Additionally, the XC4000XLA, XC4000XV and SpartanXL families also support the IDCODE instruction. The instruction codes are shown in Table 1.

Extest

While the EXTEST instruction is present in the IR, external input data and the data and 3-state control presented to the device output buffers is replaced by the content of the update latches. These latches are loaded in the Update-DR state of a data register operation of a previous SAMPLE/PRELOAD or a current EXTEST instruction (Figure 7).

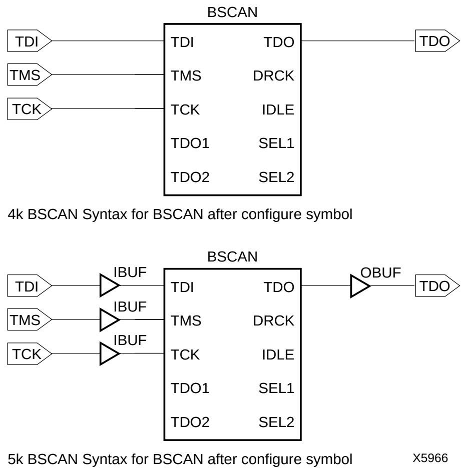


Figure 7: Boundary-scan Schematic Symbols

Since the data registers and update latches are modified during any data register operation (including BYPASS), the data in the update latches is only valid if it was loaded in the last data register cycle executed before the EXTEST line is asserted. This is corrected in the XC4000XLA, XC4000XV and SpartanXL families.

The IEEE definition of EXTEST requires that test data be driven onto outputs, that 3-state output controls be overridden, and that input data be captured. The capture of output data and 3-state controls as well as the forcing of test data into the system logic are normally performed during INTEST.

The XC4000/Spartan/XC5200 effectively perform EXTEST and INTEST simultaneously. This added functionality permits the testing of internal logic, and compensates for the absence of a separate INTEST instruction. However, when performing an EXTEST, care must be taken as to what signals are driven into the system logic. Data captured from internal system logic must be masked out of the test-data stream before performing check-sum analysis.

Sample/Preload

The SAMPLE/PRELOAD instruction permits visibility into system operation by capturing the states of the I/Os. It also permits valid data to be loaded into the test data update register prior to an EXTEST.

During SAMPLE/PRELOAD, the test data registers and update latches operate exactly as in EXTEST (see above). However, I/O operations are not affected.

Note: The boundary-scan connections for the XC4000 and Spartan families is identical.

Bypass

The BYPASS instruction permits data to be passed synchronously to the next device in a multichip boundary-scan chain. A one-bit bypass register is placed between the TDI and the TDO pins by the BYPASS instruction. The TDI input is clocked on the rising edge of TCK, the output to TDO on the falling edge.

User1, User2

These instructions permit test logic, designed by the user and implemented in CLBs, to be accessed through the TAP. Test clocks and paths to TDO are provided, together with two signals that indicate that user instructions have been loaded.

User tests depend upon CLBs and interconnects dictated by the user. Consequently, they may only be performed after configuration.

Configure

Steps to follow to configure Xilinx XC4000, Spartan, or XC5200 device via JTAG:

The serial bitstream format is identical for all configuration modes. A user can use a *design*.BIT file or a *design*.RBT file, depending on whether the user wants to read a binary file (.BIT) or an ASCII file (.RBT).

1. Enable the boundary-scan circuitry.

   This can be done one of three ways, during power-up, by configuring the device with boundary-scan enabled, or by pulling the PROGRAM pin Low.

   To enable boundary-scan during power-up, hold the INIT pin Low when power is turned on. When V_{CC} has reached V_{CC}(min), the TAP inputs can be toggled to enter JTAG instructions. The INIT pin can be held Low one of two ways, either manually or with a pull-down resistor. If you choose to manually hold INIT Low, then the INIT pin must be held low until the CONFIGURE instruction has become the current instruction. If you choose a pull-down, use a pull-down which pulls the INIT pin down to approximately 0.5V. The pull-down has the merit of automatically holding INIT Low whenever the FPGA is powered-up, and letting the user use an oscilloscope to observe relative levels of the INIT pin during configuration analysis.

2. Shift the Xilinx CONFIGURE instruction into the Instruction Register (IR). To reconfigure a device that has boundary-scan enabled after configuration, re-enter a CONFIGURE instruction.

   The Xilinx CONFIGURE instruction is 101(I_2 I_1 I_0). I_0 is the bit shifted first into the IR.

3. After shifting in the Xilinx CONFIGURE instruction, update the CONFIGURE instruction by going to the Update-IR state. When TCK goes Low in the Update-IR state, the FPGA is in the JTAG configuration mode and will start clearing the configuration memory. The CONFIGURE instruction is now the current instruction, which must be followed by a rising edge on TCK. Following update, the INIT pin may be released.

4. Following the update of the CONFIGURE instruction, the user must suppress TCK or go to the Run-Test/Idle state until the FPGA has finished clearing its configuration memory.

   The approximate time it takes to clear the FPGA configuration memory is: 2 * 1 µs * (number of frames per device bitstream). When the FPGA finishes clearing its configuration memory, the open-collector INIT goes to a high-impedance. At this point, the user should advance to the Shift-DR state. Once the TAP is in the Shift-DR state, clocks on the TCK pin will be considered configuration clocks for the serial configuration data and length count.

5. In the Shift-DR state, start shifting in the bitstream. Continue shifting in the bitstream until DONE has gone High and the startup sequence has finished.

   While in the Shift-DR state, the configuration pins LDC, HDC, INIT, PROGRAM, DOUT, and DONE all function as they normally do during non-JTAG configuration. These pins can be probed by the user. After completion of configuration, or if configuration failed, the SAMPLE/PRELOAD instruction can be used to view these IOBs (except PROGRAM and DONE.)

   LDC is Low during configuration. HDC is High during configuration. INIT will be high-impedance during configuration, but if a CRC error or frame error is detected, INIT will go Low. If a pull-down is present on INIT then the user must probe INIT with a meter or scope. With a pull-down (as in step 1) attached to the INIT pin, the user will see a drop from approximately 0.5V to 0V if INIT drops Low to indicate a data error. PROGRAM can still be used to abort the configuration process. DOUT and TDO will echo TDI until the preamble and length count are shifted into TDI. After the preamble and length count have been shifted into the FPGA, DOUT will remain High. DONE will go High when configuration is finished. Until configuration is finished, DONE will remain Low.

Additional Notes

A. The Xilinx JTAG Programmer software is available to configure the XC4000/Spartan/XC5200 devices via the boundary-scan pins.

B. It is possible to configure several FPGA devices in a JTAG chain. But unlike non-JTAG daisy-chain configuration, this does not necessarily mean merging all the bitstreams into one bitstream. In the case of JTAG configuration of Xilinx devices in a JTAG chain, all devices, except the one being configured, should be placed in BYPASS mode. The one device in CONFIGURE mode will have its bitstream downloaded to it. After configuring this device it will be placed in BYPASS, and another device will be taken out of BYPASS and into CONFIGURE.

C. If you are configuring a long daisy-chain of JTAG devices (TDI connected to TDO of the previous device), the bitstream for the device with the CONFIGURE instruction may need to have its bitstream modified.

   For example, assume that the a user has the following daisy-chain of devices:

```
source -----> device1 -----> device2 -----> device3
```

   Device1s TDO pin is connected to device2s TDI pin, and device2s TDO pin is connected to device3s TDI pin.

   The way to configure this chain is to place one device in CONFIGURE, and the other two in BYPASS. Further assume that device1 and device2 configure in this way, but device3 never configures. Specifically, device3s DONE pin never goes High. A possible cause, aside from bitstream corruption, is that the final value of the length count computed by the user/software was reached before the loading was complete.

   There are two solutions. One solution involves just continually clocking TCK (for about 16 million clocks) until DONE goes High. The other solution is to modify the bitstream; increase the length count by the number of devices ahead of the device under configuration.

   In the preceding example, the user would increase the length count value by two. (In a daisy-chain of devices configuring via boundary-scan, devices in BYPASS will supply the extra bits needed at the head of the bitstream.)

D. Once the FPGA has begun to configure, it must be completed or terminated by a 300ns or longer low pulse on the PROGRAM pin. The XC4000XLA, XC4000XV and SpartanXL families respond to a boundary-scan non-CONFIGURE instruction followed by a new CONFIGURE instruction to terminate a device configuration.

E. If boundary-scan is not included in the design being configured, then make sure that the release of I/Os is the last event in the startup sequence.

If boundary-scan is not available, the FPGA is configured, and the I/Os are released before the startup sequence is finished, the FPGA will not respond to input signals and outputs will not respond at all. This sequence requirement is not necessary in the XC4000XLA, XC4000XV and SpartanXL devices.

F. Boundary-scan configuration can fail due to a data error. Recovery in the XC4000 devices other than XC4000XLA, XC4000XV and SpartanXL devices requires assertion of the PROGRAM pin or cycling of power.

In the XC4000XLA, XC4000XV and SpartanXL families a non-CONFIGURE instruction followed by a new CONFIGURE instruction will clear the previous data and prepare for new configuration data.

G. When the CONFIGURE instruction is the current instruction, clocks on the TCK pin are not considered configuration clocks until the INIT pin has gone high impedance, and the TAP is in the Shift-DR state.

H. A n-bit instruction takes n–1 TCKs to load into a n-bit register. In the IEEE 1149.1 implementation of the TAP, a shift in the IR/DR happens whenever the TAP is in the Shift-IR or Shift-DR state, as well as when leaving the Shift-IR / Shift-DR state. Based on this behavior, it takes n–1 TCKs to shift in a n-bit instruction. In the case of the XC4000/Spartan/5200 devices, the user only needs to stay in the Shift-IR state for two TCKs to shift in a 3-bit instruction.

I. If the user is attempting to configure a chain of devices, it is recommended that the user only configure the chain in all boundary-scan mode, or use a non-boundary-scan configuration mode. While it is possible to configure a daisy-chain of devices, some in boundary-scan and some in non-boundary-scan configuration, this will not necessarily give the user a continuous boundary-scan chain. This may or may not be a problem, depending on the specific application.

J. Configuring a chain of Xilinx FPGAs via boundary-scan does not require merging all the bitstreams into one prom file, as in non-boundary-scan configuration daisy-chains. However, the same configuration circuitry is used for both boundary-scan and non-boundary-scan configuration. Thus, if a user would like, it is possible to merge all bitstreams into one, using MakePROM/promgen or the PROM File Formatter. In a case where the user wants to merge the bitstreams into one prom file, the user should configure as in (B) above. Additionally, the user will have to tie all INIT pins together. All DONE pins could also be tied together.

K. In the XC4000XLA, XC4000XV and SpartanXL parts, the status of a CONFIGURE operation can be verified by testing the capture values of bits 0 (TDO.T) and 1 (TDO.O) from a SAMPLE/PRELOAD operation (Table 3). For a valid successful status the bitgen option of BSCAN_Status must be enabled (default disable).

Table 3: Status Bits For a Configure Operation

Bit 0 (TDO.T)	BIT 1 (TDO.O)	
0	0	Unsuccessful, no data error
1	0	Unsuccessful, data error
0	1	Successful

Readback

Readback through boundary-scan allows the user to access the readback features of the device, which would normally need to be accessed through user-specified pins. Readback through the TAP follows the same guidelines that non-boundary-scan readback follows. In both cases, readback can be accomplished at a minimum of 1 MHz and a maximum of 2 MHz. Additionally, the readback bitstream through boundary-scan has the same format as non-boundary-scan readback. Loading the IR with the READBACK instruction and then advancing to the Update-IR state initializes the readback process. Data is clocked out of TDO in the Shift-DR state, clocked by TCK.

Unlike regular readback, which can be done repeatedly, readback through the TAP requires the following circuit:

1. In your schematic, or top-level synthesis design, instantiate the BSCAN and READBACK symbols.
2. Connect the BSCAN symbol pins TDI, TMS, TCK, and TDO to the boundary-scan pads TDI, TMS, TCK, and TDO, respectively.
3. Next, connect the net between the TCK pad and TCK pin on the BSCAN symbol to an IBUF. Take the output of the IBUF and connect it to the CLK pin of the READBACK symbol. See Figure 8.

   For the XC5200, the equivalent circuit must be implemented using the XACT Design Editor (XDE) program EditLCA, or FPGA Editor in the Alliance/Foundation tools.

   Using XDE: After placing and routing your XC5200 design, load the *design*.LCA file into EditLCA, and follow the procedures below: (<ENTER> means hit the enter key on your keyboard)

   A. Once EditLCA has displayed the *design*.LCA file, type the following:

   ```
   eb bscan <ENTER>
   ```

   This will bring up the Editblock window for the XC5200 BSCAN symbol.

   B. In the Editblock window, select the ’used’ option, which is in the upper left corner of the screen.

   C. Now type:

   ```
   endb <ENTER>
   ```

   This brings you back to the EditLCA screen.

   D. Next type the following:

   ```
   addnet username tckpin.i rdbk.ck <ENTER>
   ```

where tckpin is the pin number of the TCK pin of your XC5200 device. ’username’ is a net name of your choIce. For example, if your design used an XC5202PC84, then the above command line would be:

```
addnet mynet p16.i rdbk.ck <ENTER>
```

E. At this point you should see a net go from the TCK pin to the CK pin of the Readback symbol.

F. Save your changes to the LCA file and exit XDE

G. After entering the above circuit, compile the design to an LCA file.

H. Make the bitstream file for the LCA file by using the following option with: `makebits, -f readclk:rdbk.`

For example at a unix prompt, type:

```
%makebits -f readclk:rdbk design
```

Using FPGA Editor: Load the placed and routed design.NCD file into FPGA Editor in the Read/Write mode.

A. Once FPGA Editor displays the design.NCD file, use the FIND command to find the sites RDBK and P16 (For a XC5202PC84, this would be different for a different package. Find the correct pad location for the TCK pin from the databook).

B. Select each component by clicking on them once. Then use the Edit -> Add menu commands to add them in.

C. Select the pins named RDBK.CK and TCK.I on those components respectively and click on the AUTOROUTE button. You should see a net going from TCK.I pin to the RDBK.CK pin.

D. Save your changes to the NCD and PCF files and exit FPGA Editor.

E. Create the bitstream from the NCD file by using the following option with `bitgen, -g ReadClk:Rdbk.`

For example at a unix prompt, type:

```
bitgen -g ReadClk:Rdbk design.ncd output_file pcffile.pcf
```

4. Now the FPGA is ready to perform consecutive readbacks.

   Readback is performed by loading the IR with the READBACK instruction and then shifting out the captured data from the Shift-DR state in the TAP. Readback data is captured when READBACK is made the current instruction in the TAP.

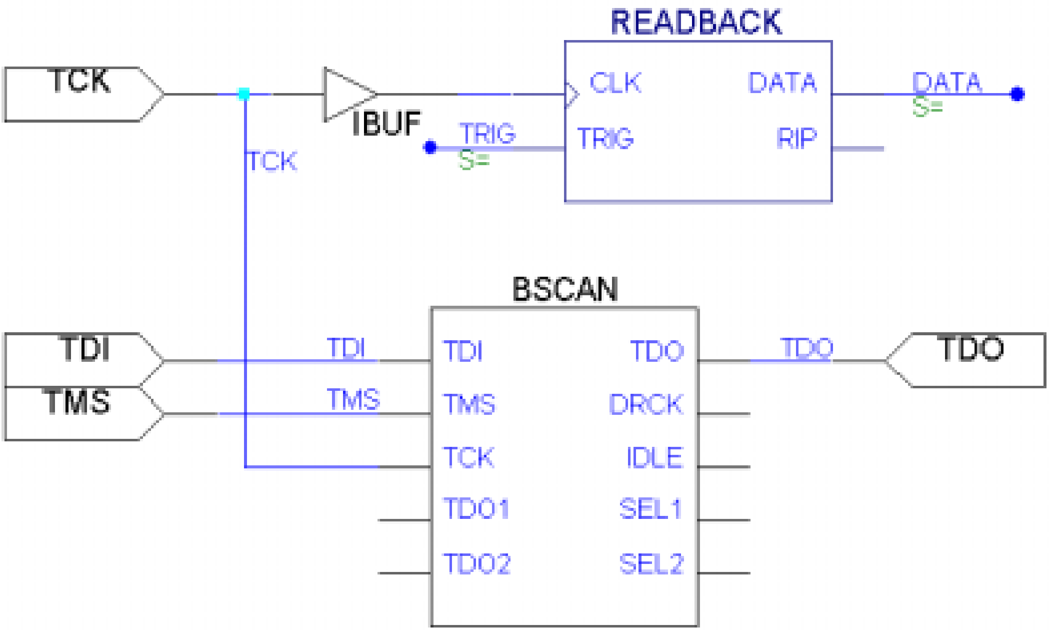
   Perform the first readback by loading the IR with the READBACK instruction. This first readback must be finished, which means shifting out the "entire" readback bitstream. To be safe, shift out the entire bitstream and then send three additional TCKs.

5. After performing the first readback, another readback can be performed by going to the Test-Logic-Reset state, and reloading the READBACK instruction and performing the Readback as described in the previous paragraph.

In summary, consecutive readbacks are performed by starting from Test-Logic-Reset, loading the IR with the READBACK instruction, shifting out the readback bitstream plus three additional TCKs, and then going back to the Test-Logic-Reset state.

Alternatively, it is possible to perform consecutive readbacks without returning to the Test-Logic-Reset state. Realize that after shifting out the readback bitstream, a minimum of three additional clocks are needed on the readback register. Thus, after doing a readback, instead of going back to Test-Logic-Reset, a user can opt to execute some other JTAG instruction, and then perform another readback.

This procedure is only needed if more than one readback is desired. If readback is only to be performed once, then the connection between the BSCAN symbol and the READBACK symbol is not needed. In that case, all that is needed is the BSCAN symbol instantiated with the boundary-scan pads (TDI, TMS, TCK, and TDO) on the top level of the design.



Notes:

* For the 5K devices you must use ibuf/obuf in addition to TAP Pads.

** The S = attribute is just one way to keep the implementation tools from optimizing out the readback

Figure 8: Symbol Connectivity for Performing Multiple Readbacks Through the Jtag Port

Boundary-scan Description Language Files

Boundary-scan Description Language (BSDL) files describe boundary-scan-capable parts in a standard format used by automated test-generation software. The order and function of bits in the boundary-scan data register are included in this description.

BSDL files are available from the Xilinx Support web site at http://support.xilinx.com.

Bibliography

The following publications contain information about the IEEE Standard 1149.1, and should be consulted for general boundary-scan information beyond the scope of this application note.

Colin M. Maunder & Rodham E. Tulloss. *The Test Access Port and Boundary-scan Architecture*. IEEE Computer Society Press, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1264. See www.computer.org/cspress/catalog/st01096.htm

John Fluke Mfg. Co. Inc. *The ABC of Boundary-scan Test*. John Fluke Mfg. Co. Inc., P.O. Box 9090, Everett, WA 98206.

GenRad Inc. *Meeting the Challenge of Boundary-scan*. GenRad Inc., 300 Baker Ave., Concord, MA 01742-2174.

Ken Parker. *The Boundary-scan Handbook*. Kluwer Academic Publications, (617) 871-6600.

IEEE Standards, standards.ieee.org

Texas Instruments, www.ti.com/sc/docs/jtag/jtaghome.htm

Revision History

Date	Version #	Revision
11.16.99	3.0	Reformatted to one-column and updated.