XILINX®

Boundary Scan in XC4000 and XC5200 Series Devices

XAPP017 December 10, 1997 (Version 2.1)

Application Note

Summary

XC4000 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.

Xilinx Family

XC4000 Series, XC5200

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 input 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 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 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. To test the external interconnect, devices drive values onto their outputs and observe input values received from other devices. A central test controller compares the received data with expected results. Data to be driven onto outputs is distributed through a chain of shift registers, and observed input data is returned through the same shift-register path.

Data is passed serially from one device to the next, thus forming a boundary-scan path or loop that originates at the test controller and returns there. 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/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/XC5200 Boundary-Scan Features

XC4000/XC5200 devices support all the mandatory boundary-scan instructions specified in the IEEE Standard 1149.1. A Test Access Port (TAP) and registers are provided that implement the EXTEST, SAMPLE/PRELOAD and BYPASS instructions. The TAP can also support two USERCODE instructions.

Note**: If boun**dary scan is not used after the device is configured, the user can use the special boundary scan pads as input or output pins. During configuration, be sure not to toggle the TAP pins, since inadvertent toggling of the TAP pins can turn the boundary scan circuitry 'on.' The TDI, TMS, and TCK pads can be used as unrestricted I/O. The TDO pad can be used as an output pad. In the XC5200 family, all four pins have full I/O capability. And like the regular IOBs, these input and output pins have pullups and pulldowns available.

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

Additionally, internal signals can be captured during EXTEST by connecting them to unbonded IOBs, or to the unused outputs in IOBs used as unidirectional input pins. This partially compensates for the lack of INTEST support.

The public boundary-scan instructions are always available prior to configuration. After configuration, the public instructions and any USER1/USER2 instructions are only available if boundary scan specified in the schematic/HDL code. While SAMPLE and BYPASS are available during configuration, it is recommended that boundary-scan operations not be performed during this transitory period.

In addition to the test instructions outlined above, the boundary-scan circuitry can also be used to configure the FPGA device, and read back the configuration data.

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

Deviations from the IEEE Standard

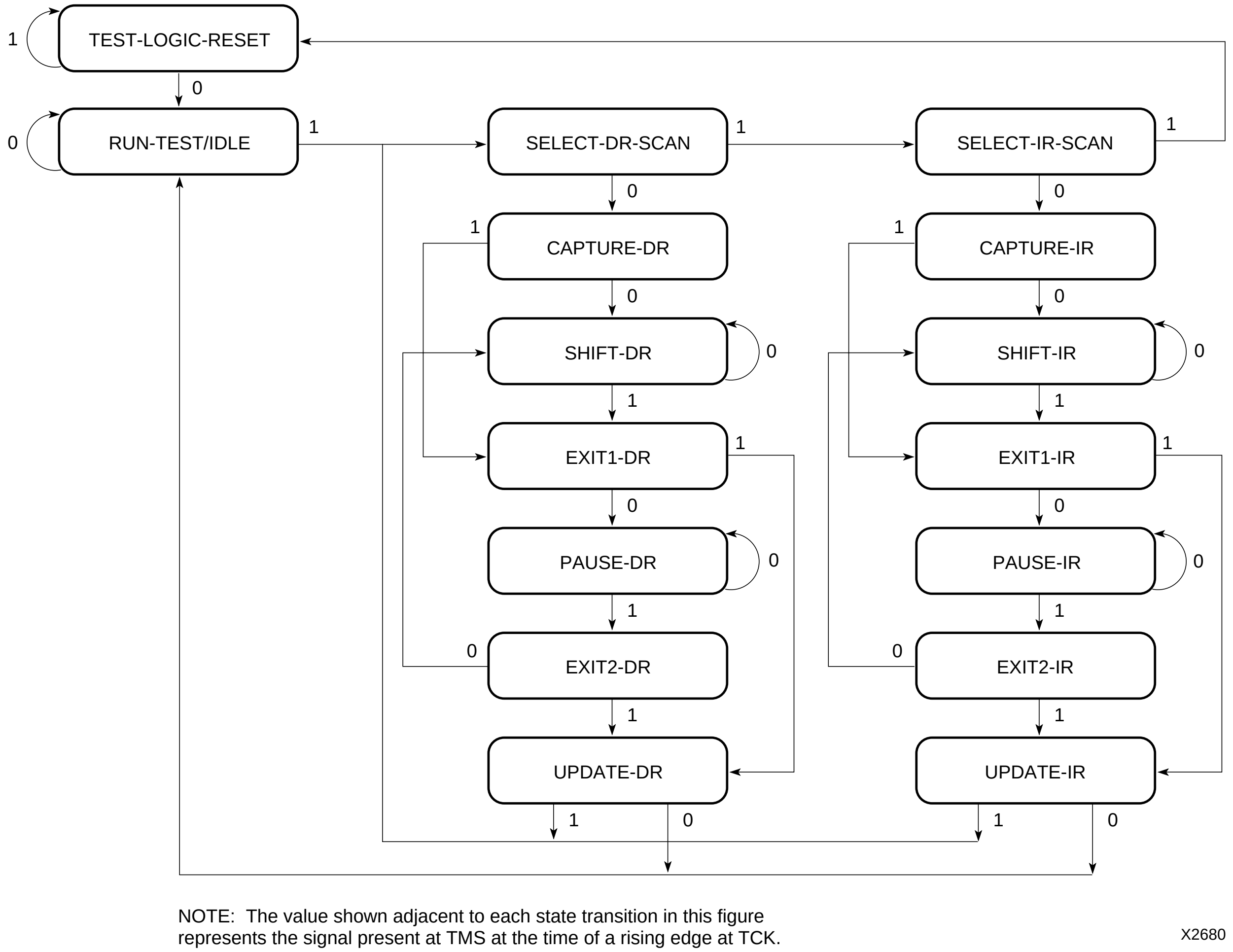
The XC4000/XC5200 boundary scan implementation deviates from the IEEE standard in that three dedicated pins (CCLK, $\overline{\text{PROGRAM}}$ and DONE) are not scanned.

It should also be noted that the Test Data Register contains three Xilinx test bits (BSCANT.UPD, TDO.O and TDO.T) and that 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, and system clock inputs are not disabled during EXTEST, as recommended in the standard.

The TAP pins (TMS, TCK, TDI and TDO) are scanned, but connections to the TAP controller are made before the boundary-scan logic. Consequently, the operation of the TAP controller cannot be affected by boundary-scan test data.

When the TAP is in the shift-DR state the contents of all data registers are shifted; if you are in the middle of shifting out data from the data register, complete shifting out of all data first, before switching to the instruction or bypass 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.

Figure 1: State Diagram for the TAP Controller

Boundary-Scan Hardware

Test Access Port

The boundary-scan logic is accessed through the Test Access Port (TAP), which comprises 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.

The TAP pins are permanently connected to the boundary-scan circuitry. However, once the device is configured, the connections may be ignored unless the use of boundary scan is specified in the design. See “Using Boundary Scan” on page 13-57.

If the use of boundary scan is specified, the TAP input pins (TMS, TCK and TDI) may still be shared with other logic, subject to limitations imposed by external connections and the operation of the TAP Controller. In designs that do not use boundary scan after configuration, the TAP pins can be used as inputs or outputs from the user logic in the FPGA device. TMS, TCK and TDI are available as unrestricted I/Os, while TDO only provides a 3-state output. In the XC5200 family, all four pins are available as I/O.

Before the FPGA is configured, it is important not to toggle the TAP pins (TDI, TMS, TCK), since these pins ‘turn-on’ boundary scan. Before an FPGA is configured, at a minimum, do not toggle TCK. Similarly, if boundary scan is enabled in a design after the FPGA is configured, care must be taken not to toggle the TAP pins (TDI, TMS, TCK) to prevent turning ‘on’ boundary scan by accident.

TAP Controller

The TAP Controller is a 16-state machine that controls the operation of the boundary-scan circuitry in response to TMS. This state machine implements the state diagram specified by the IEEE standard (Figure 1) and is clocked by TCK.

Upon power-on, or if the boundary scan logic is not used in the application, the TAP controller is forced into the Test-Logic-Reset state. After configuration, the controller remains disabled, unless its use is explicitly specified in the user design. $\overline{\text{PROGRAM}}$ resets the latched decodes for EXTEST, CONFIGURE, and READBACK instructions.

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 device input and output data (boundary-scan 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
1	0	1	CONFIGURE	DOUT	Disabled
1	1	0	RESERVED	—	—
1	1	1	BYPASS	Bypass Register	—

I_0 is closest to TDO

Note: Whenever the TAP Controller is in the Shift-DR state, all data registers are shifted, regardless of the instruction. DR data is modified even if a BYPASS instruction is executed.

The instruction register is used not only to hold the current instruction. If the TAP is in the capture-IR state and TCK goes high, the instruction register captures the current boundary-scan state of the device. I_0 is 1 by default. I_1 is 0 by default. I_2 is 0 if the device is in configure by boundary scan mode. Before and after configure by boundary scan mode, I_2 will capture 1. Note that I_0 is shifted out of TDO first, then I_1, and then I_2.

The Boundary-Scan Data Register

The Data Register (DR) is a serial shift register implemented in the IOBs of the FPGA device, (Figure 2). Potentially, each IOB can be configured as an independently controlled bidirectional pin. Therefore, three data register bits are provided per IOB: for input data, output data and 3-state control. In practice, many of these bits are redundant, but they are not removed from the scan chain.

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

In a typical DR instruction, the DR captures data during the Capture-DR state (on the rising edge of TCK). This data is then shifted out and replaced with new test data. 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 an EXTEST instruction is in progress.

Note: The update latch is opened whenever the TAP Controller is in the Update-DR state, regardless of the instruction. Care must be exercised 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.

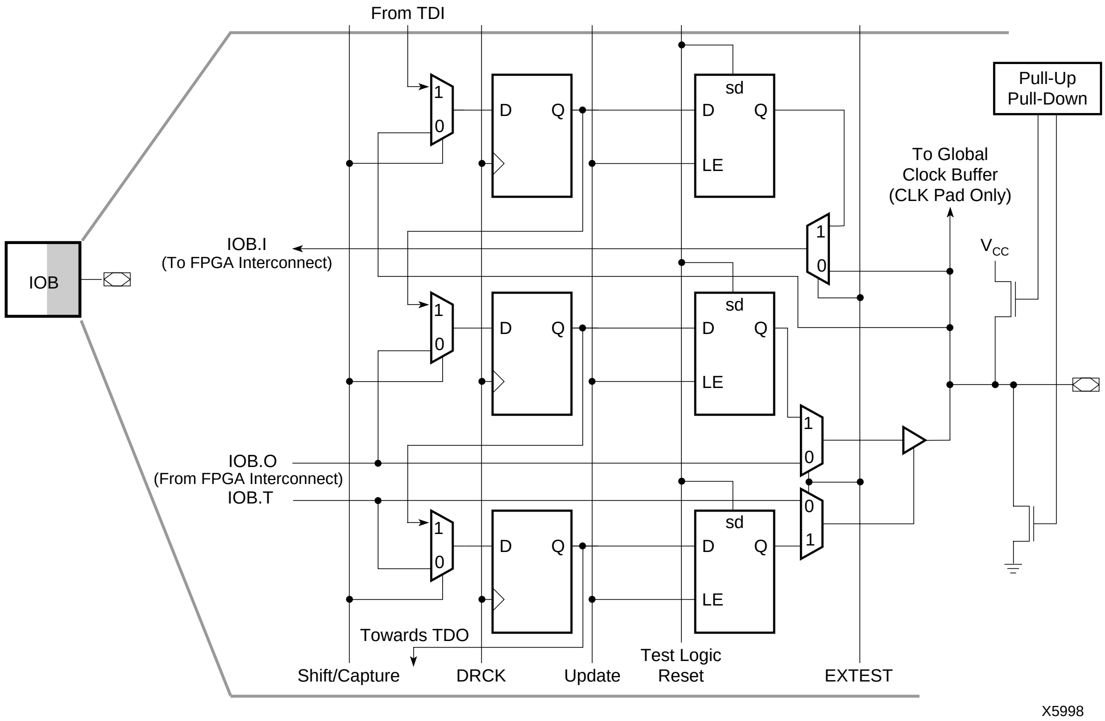


Figure 2: Boundary Scan Logic in a Typical IOB

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 helps compensate for the lack of INTEST. Logic inputs may be set to specific levels by a SAMPLE/PRELOAD or EXTEST instruction and the resulting logic outputs captured during a subsequent EXTEST. It must be recognized, however, that all DR bits are captured during an EXTEST and, therefore, may change.

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 PC traces.

The primary and secondary global clock inputs (PGCK1-4 and SGCK1-4 in XC4000, GCK1-4 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

Table 2 lists, in data-stream order, the boundary-scan cells that make up the DR 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. This order is consistent with the BSDL description.

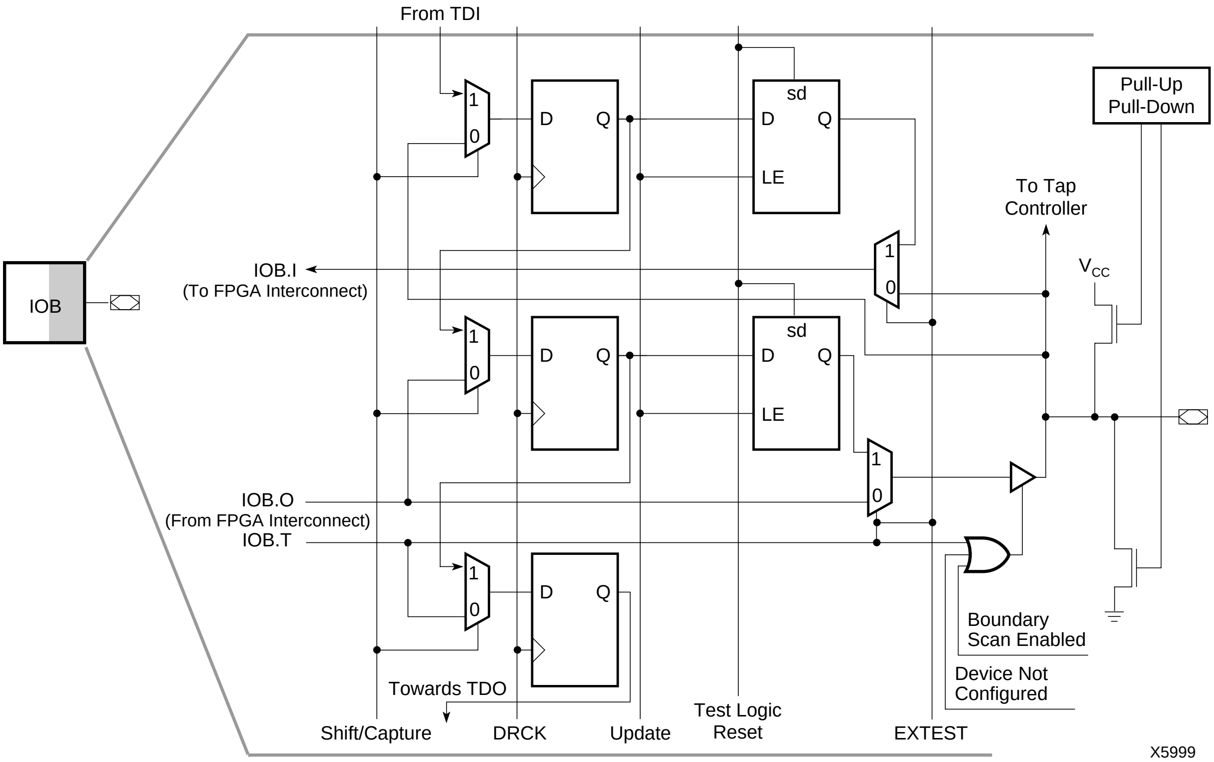


Figure 3: Boundary Scan Logic in a TAP Input (TMS, TCK, and TDI Only)

Each IOB corresponds to three bits in the DR. 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 EPIC. 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: 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

Note: All IOBs remain in the DR, 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. CCLK, $\overline{\text{PROGRAM}}$ and DONE are not included in the boundary scan.

Tables in the data sheets show the DR order for all XC4000/XC5200 family devices. The DR also includes the following non-pin bits: TDO.T and TDO.I, which are always bits 0 and 1 of the DR, respectively, and BSCANT.UPD which is always the last bit of the DR.

The Bypass Register

This is a 1-bit shift register that passes the serial data directly to TDO when a BYPASS instruction is executed.

User Registers

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

Note: The TDI signal supplied to user test logic is overwritten by boundary-scan test data during EXTEST. During user 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 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. It is provided to clock user test-data registers. TDI data should be sampled with the falling edge of DRCK (rising edge of TCK). The TDO output flip-flop accepts data on the rising edge of DRCK (falling edge of TCK). DRCK is active only during the Capture-DR and Shift-DR states of the TAP controller. When not active in the XC4000, DRCK is Low. In the XC5200, when DRCK is not active, it is High.

IDLE – IDLE is a second gated and inverted 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 pin 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 pin 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.

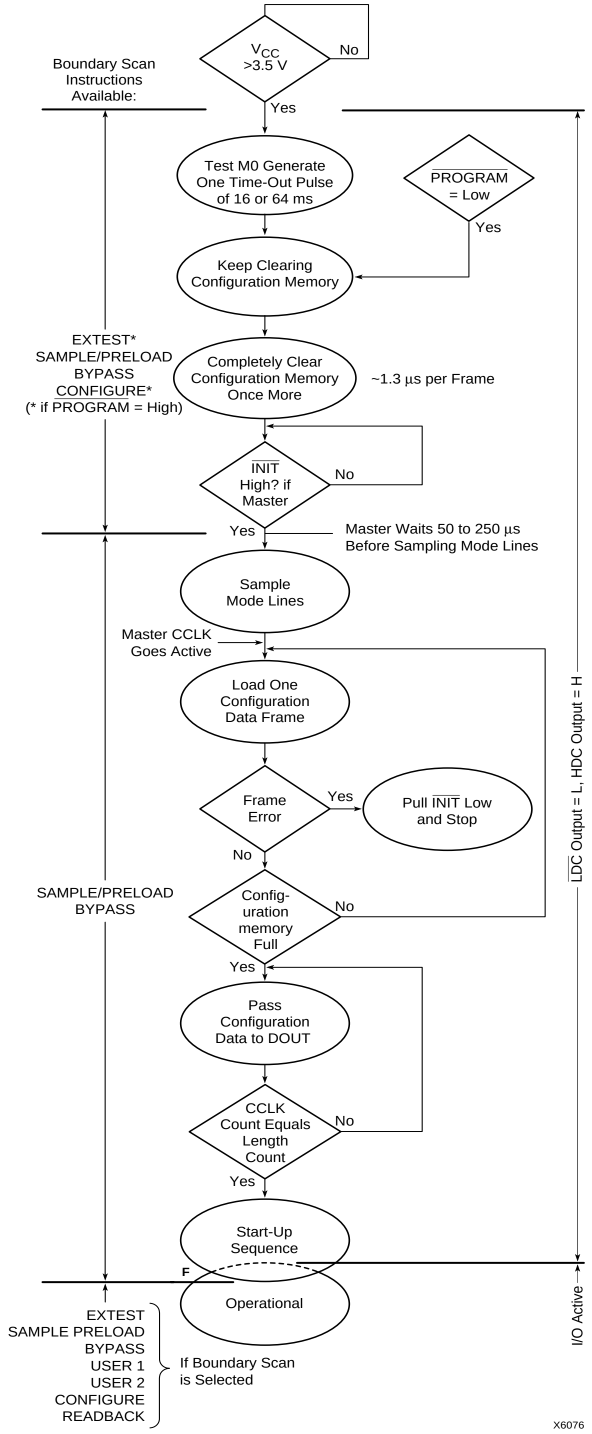
SHIFT - This pin is only available in the XC5200 boundary scan symbol. When the USER1 or USER2 instructions are used, SHIFT is High, in all other cases SHIFT is Low.

Using Boundary Scan

Full access to the built-in boundary-scan logic is always available between power-up and the start of configuration. Optionally, the built-in logic is fully available after configuration if boundary scan is specified in the design. At this time, user test logic is also available, and may be accessed through the boundary-scan port. During configuration, a reduced boundary-scan capability remains available: the SAMPLE/PRELOAD and BYPASS instructions only.

Figure 4 is a flow chart of the XC4000 FPGA start-up sequence that shows when the boundary-scan instructions are available. Since PROGRAM resets the TAP controller, boundary-scan operations cannot commence until PROGRAM has been taken High.

.



V_{CC} >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
EXTEST
SAMPLE PRELOAD
BYPASS
USER 1
USER 2
CONFIGURE
READBACK
If Boundary Scan is Selected
LDC Output = L, HDC Output = H
I/O Active
X6076

Figure 4: XC4000 Start-up Sequence

Full boundary-scan capabilities are available until INIT is High. Without external intervention, INIT automatically goes High after ~1 ms. If more time is required for boundary-scan testing, INIT may be held Low beyond this period by applying an external Low signal to the INIT pin until testing is complete. Once INIT has gone High, all clocks on the TCK pin are counted as configuration clocks for data and length count. See “CONFIGURE” on page 13-59. for more details.

Boundary scan can be accessed before the FPGA is configured and after the FPGA is configured. If you want to access boundary scan before the device is configured, then when you power-up the device, hold the INIT pin Low until V_{CC} has risen to V_{CC}(min).

If you have already started configuring the device, and data frames are already being sent to the FPGA, then you have two choices. You can either access full-boundary scan mode, or limited boundary scan mode. If you want to access full-boundary scan mode, then both INIT and PROGRAM must be brought Low (Hold INIT and PROG Low for over 300 ns and then release PROGRAM.) After releasing PROGRAM, continue to hold INIT Low while sending signals to the TAP. If you can use the limited boundary scan mode (which means you only can use the SAMPLE/PRELOAD and BYPASS instructions), then just bring INIT Low.

Accessing boundary scan after the device is configured has one requirement. The BSCAN symbol must be instantiated/inserted into your design with the correct syntax (see Figure 5). In this case, activating boundary scan after configuration amounts to toggling the TAP pins.

.

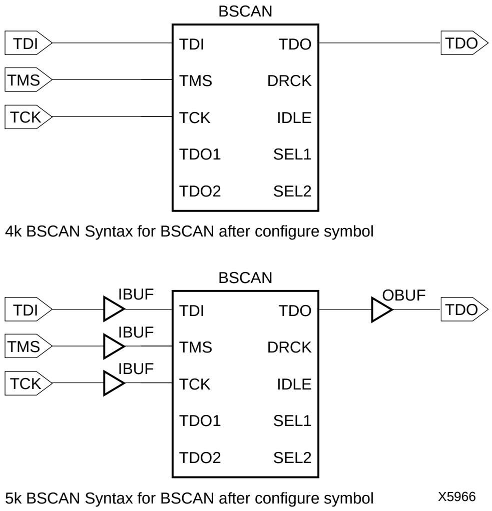


Figure 5: Boundary-Scan Schematic Symbols

If the BSCAN symbol is not included, 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 6.)

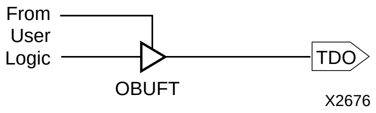


Figure 6: Typical Non-Boundary-Scan TDO Connection

Boundary Scan Instructions

The XC4000/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). The instruction codes are shown in See Table 1 on page 13-54.

EXTEST

While the EXTEST instruction is present in the IR, the data presented to the device output buffers is replaced by data previously loaded through the boundary-scan DR and stored in the update latch (Figure 7). SImilarly, the output 3-state controls are replaced, and the data passed to internal system logic from input pins is replaced.

When a DR instruction cycle is executed, data arriving at the device input pins is loaded into the DR. The data from the system logic that drives output buffers and their 3-state controls is also loaded. This action occurs during the CAPTURE-DR state of the TAP controller (Figure 1 on page 13-53). Data is serially shifted out of the DR during the SHIFT-DR state; simultaneously, new data is shifted in. In the UPDATE-DR state, the new data is transferred into the update latch for use as replacement data, as described above.

The replacement of system data with update latch data starts as soon as the EXTEST instruction is loaded into the IR. For this data to be valid, it must have been loaded by a previous EXTEST or SAMPLE/PRELOAD operation.

Since the DR and update latch are modified during any DR instruction cycle, including BYPASS, the data in the update latch is only valid if it was loaded in the last DR instruction cycle executed before EXTEST is asserted.

The IEEE definition of EXTEST only 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 and the forcing of test data into the system logic is normally performed during INTEST.

The XC4000/XC5200 effectively performs 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 sig-

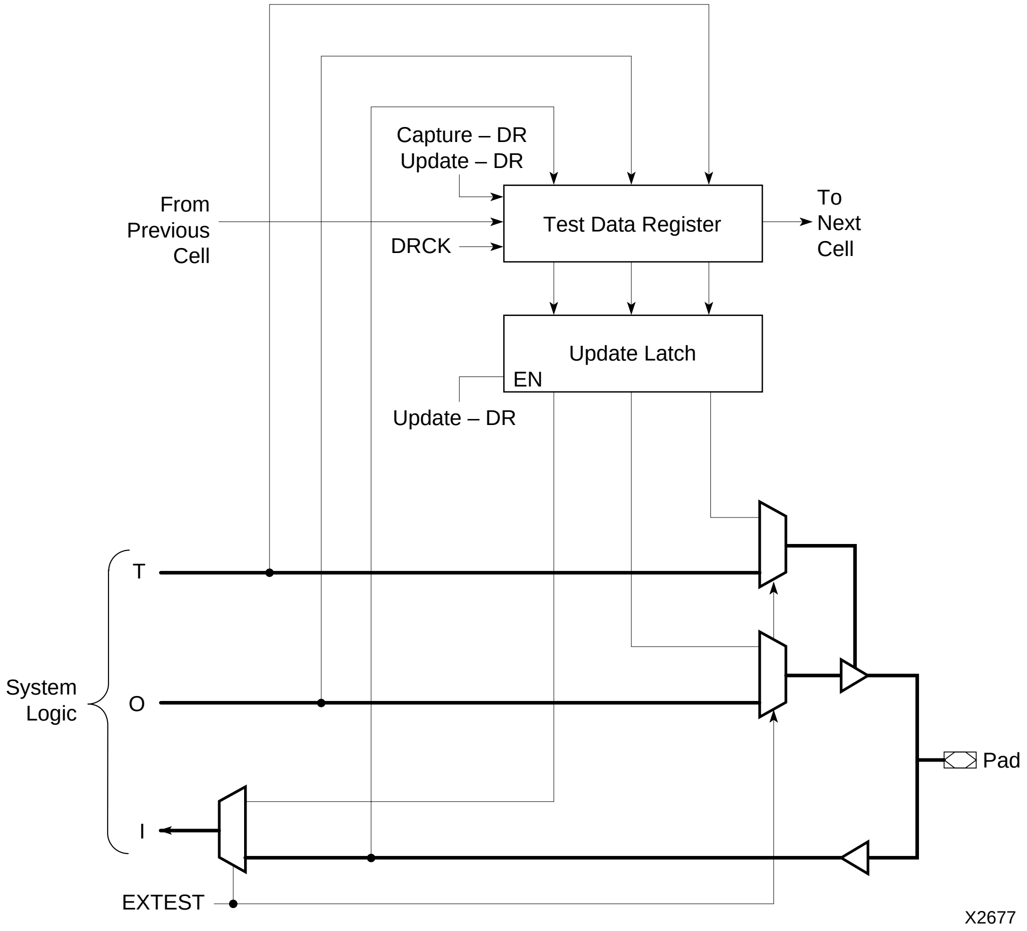


Figure 7: EXTEST Data Flow

nals 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 state of the I/O. It also permits valid data to be loaded into the update register before commencing an EXTEST.

The DR and update latch operate exactly as in EXTEST (see above). However, data flows through the I/O unmodified.

BYPASS

The BYPASS instruction permits data to be passed synchronously to the next device in the boundary-scan path. There is a 1-bit shift register between the TDI and TDO flip-flop.

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. See “User Registers” on page 13-56.

User tests depend upon CLBs and interconnect that must be configured to operate. Consequently, they may only be performed after configuration.

CONFIGURE

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

The 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, either during power-up, or by configuring the device with boundary scan enabled, or by pulling the $\overline{\text{PROGRAM}}$ pin low.

   To enable boundary scan during power-up, hold the $\overline{\text{INIT}}$ 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 $\overline{\text{INIT}}$ pin can be held Low one of two ways, either manually or with a pulldown. If you choose to manually hold the $\overline{\text{INIT}}$ low, then the $\overline{\text{INIT}}$ pin must be held low until the CONFIGURE instruction is the current instruction. If you choose a pulldown, use a pulldown which pulls the $\overline{\text{INIT}}$ pin down to approximately 0.5V. The pulldown has the merit of holding $\overline{\text{INIT}}$ low whenever the FPGA is powered-up, and letting the user observe the $\overline{\text{INIT}}$ pin during configuration.

After the FPGA has been configured, if you want to reconfigure a configured device that has boundary scan enabled after configuration, then just start toggling the boundary scan TAP pins.

2. Load the Xilinx CONFIGURE instruction into the Instruction Register (IR).

   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, make the CONFIGURE instruction the current JTAG instruction by going to the UPDATE-IR state. When TCK goes low in the UPDATE-IR state, the FPGA is now 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. If you chose to manually hold the $\overline{\text{INIT}}$ pin Low, then the $\overline{\text{INIT}}$ pin must be held Low until the CONFIGURE instruction is the current instruction.

4. Once the Xilinx CONFIGURE instruction has been made the current instruction, the user must go to the RUN-TEST/IDLE state, and remain in 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 us * (number of frames per device bitstream).

   When the FPGA has finished clearing its configuration memory, the open-collector $\overline{\text{INIT}}$ has gone high impedance. At this point, the user should advance to the SHIFT-DR state. Once the TAP is in the SHIFT-DR state and the $\overline{\text{INIT}}$ pin has been released, clocks on the TCK pin will be considered configuration clocks for 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.

   During the time you are shifting in the bitstream via the TAP, the configuration pins $\overline{\text{LDC}}$, HDC, $\overline{\text{INIT}}$, $\overline{\text{PROGRAM}}$, 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 $\overline{\text{PROGRAM}}$ and DONE.)

   $\overline{\text{LDC}}$ is Low during configuration. HDC is High during configuration. $\overline{\text{INIT}}$ will be high impedance during configuration, but if a CRC error or frame error is detected, $\overline{\text{INIT}}$ will go Low. If a pulldown is present on $\overline{\text{INIT}}$ then the user must probe $\overline{\text{INIT}}$ with a meter or scope. With a pulldown (as in step 1) attached to the $\overline{\text{INIT}}$ pin, the user will see a drop from approximately 0.5V to 0V if $\overline{\text{INIT}}$ drops Low to indicate a data error. $\overline{\text{PROGRAM}}$ 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) It is possible to configure several XC4000/XC5200 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, will 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 into CONFIGURE.

(b) 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

Device1's TDO pin is connected to device2's TDI pin, and device2's TDO pin is connected to device3's 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, device3's DONE pin never goes High. The problem is the bitstream length count. 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 15 seconds) 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 2. (In a daisy-chain of devices configuring via boundary scan, devices in BYPASS will supply the extra 1s needed at the head of the bitstream.)

(c) In general for the XC4000 and XC5200, if you are configuring these devices via JTAG, finish configuring the device first before executing any other JTAG instructions. Once configuration through boundary scan is started, the configuration operation must be finished.

(d) 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.

(e) Re-issuing a boundary scan CONFIGURE instruction after the clearing of configuration memory will cancel the CONFIGURE instruction.

The proper method of re-issuing a CONFIGURE instruction after the configuration memory is cleared is to issue another boundary scan instruction, and follow it by the CONFIGURE instruction.

(f) If configuration through boundary scan fails, there are only two boundary scan instructions available: SAMPLE/PRELOAD and BYPASS. If another reconfiguration is to be attempted, then the PROGRAM pin must be pulled Low, or the FPGA must be repowered.

(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) 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 the non-boundary scan configuration modes. It is possible to configure a daisy-chain of devices, some in boundary scan and some in non-boundary scan configuration. Configuring in a mixed mode will not necessarily give the user a continuous boundary scan chain, which may or may not be a problem for a particular user's applications.

(j) Currently, there is no software to configure a Xilinx FPGA via the boundary scan pins. The user must provide this.

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

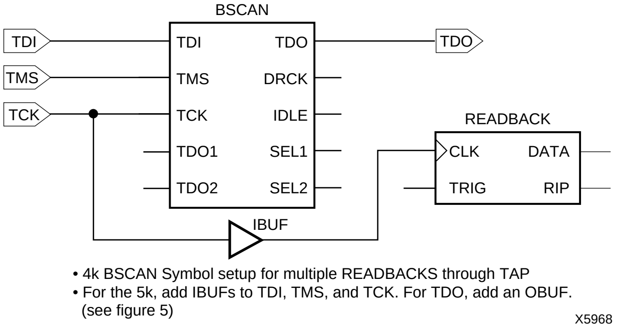
NOTE: The intention of configuration for a daisy-chain was to use either all the devices in boundary scan, or all the devices in non-boundary scan configuration.

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. All limits of ‘normal’ readback are the same with readback through the TAP. Like regular readback, readback through the TAP is at a minimum of 100 KHz and at a maximum of 2 MHz. Like regular readback, the readback bitstream through boundary scan has the same format.

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.



- 4k BSCAN Symbol setup for multiple READBACKS through TAP
- For the 5k, add IBUFs to TDI, TMS, and TCK. For TDO, add an OBUF. (see figure 5)

Figure 8: Symbol Setup for Multiple Readbacks

For the XC5200, the equivalent circuit must be implemented using the XACT Design Editor (XDE) program EditLCA, or EPIC in the M1-based tools. 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.

4. After entering the above circuit, compile the design to an LCA file.
5. Make the bitstream file for the LCA file by using the following option with makebits, or use the M1 Bitstream Generator:

   -f readclk:rdbk
   For example, at a unix prompt:
   % makebits -f readclk:rdbk *design*
6. 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.
7. After performing the first readback, another readback can be performed by going to the TEST-LOGIC-RESET state, and re-loading 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, if you do not want to go back 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. So, 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.

Also, this procedure is only needed if you intend to do more than one readback. If you intend only do a readback 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, & TDO) on the top level of the design.

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 in the Xilinx File Download area via the Xilinx WebLINX web site (www.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