

Design For Test (DFT)

Guidelines for Boundary-Scan Testing

A Guide for PCB Designers, Test Engineers and Managers

Copyright © 1997 – 2011, Corelis Inc.

Corelis, Inc.
13100 Alondra Blvd.
Cerritos, CA 90703 USA
Tel.: +1 562-926-6727 • Fax: +1 562-404-6196

Preface

GENERAL NOTICE

Information contained in this document is subject to change without notice. CORELIS shall not be liable for errors contained herein for incidental or consequential damages in connection with the furnishing, performance, or use of material contained in this manual.

Table of Contents

Introduction	1
Why Test	1
What is Boundary-Scan	1
What is JTAG Emulation Test?	3
Design Considerations	4
Real-Estate Requirements	4
Component Selection	6
Off-The-Shelf Devices	6
Custom Devices (ASIC)	8
Boundary-Scan Chain	10
Connectors, Signals and Termination	10
Boundary-Scan PCB-Routing Considerations	13
Group Components with Similar Logic Characteristics.....	13
Number of TAPs	13
Signal Buffering.....	14
Bypassing.....	15
Avoiding Problems with the TAP Signals	17
Configurable BSDs (Boundary-Scan Devices)	18
Compliance Enable Pins	18
Testing with Programmed and Configured Devices	24
General Configuration for Xilinx INIT and Altera nCONFIG.....	27
Control of Non-Boundary-Scan Devices	30
Testing Memory Devices using Memory Clusters.....	30
Clock-Driven Memory SDRAM DDR	32
Flash Programming	33
Connectors and Other Interfaces	37
Device-Specific Considerations	38
Test Expectations	38
At-Speed Functional Testing	39
Additional Testing	40
Finding Faults: Now What?	41
Appendix A: Glossary of Abbreviations	42
Appendix B: Device-Specific Requirements	44
Actel.....	44
Altera	45
Freescale / Motorola	46
IDT.....	52
Intel	52
Lattice	52
Lucent	54

PLX Technology	54
Texas Instruments	55
Xilinx	56
Vantis	57
And other devices that do not include a boundary-scan register	57
Multi-Chip Modules.....	57
<i>Contact information</i>	58

Table of Figures

Figure 1: Block diagram of a Single Boundary-Scan Device (BSD)	2
Figure 2: Typical Connection of BSDs on a Target Board.....	4
Figure 3: Target Board with Controller and Power Connected	5
Figure 4: Incorporated USB2JTAG Controller on Board.....	6
Figure 5: 10-pin TAP Connector Schematic.....	10
Figure 6: 16-pin TAP Connector Schematic.....	11
Figure 7: 20-pin TAP Connector Schematic for SPI	12
Figure 8: Bypassing manually with jumpers	15
Figure 9: Bypassing with added logic.....	16
Figure 10: Do Not Connect TRST* to I/O Pins of Devices in the Scan Chain	17
Figure 11: Do Not Connect the TAP Signals to I/O Pins of Devices in the Scan Chain..	17
Figure 12: Do Not Connect Compliance Enable Pins to Other Devices in the Chain.....	21
Figure 13: Forcing Compliance Enable Pins for Boundary-scan Testing	22
Figure 14: Memory Devices Surrounded by Boundary-Scan Devices.....	31
Figure 15: SDRAM clock driver is 'free running'.....	32
Figure 16: SDRAM clock controlled by 2 boundary-scan signals	32
Figure 17: Flash Programming Speed is Limited by the Slowest Device in the Chain....	35
Figure 18: Separate the Scan Chains to Increase Flash Programming Speed.....	35
Figure 19: Implementation of a Scan I/O Module	37
Figure 20: Flying Prober and BON testers	40
Figure 21: Corelis ScanExpress Viewer showing isolated faults	41
Figure 22: One scan chain using multiple ScanPath Linker.....	53
Figure 23: Another Sample of ScanPath Linker	54

Introduction

In today's fast paced environment with short time-to-market requirements, it has become increasingly important to design products that allow for early fault and failure detection. The earlier a mistake or a defect can be detected in the design phase or in the production process, the less money it will cost to remedy it and the sooner the product will be ready for production or shipment. Therefore, a good Design-For-Test (DFT) strategy is needed for the design, prototype and production phase of a product.

In this document, DFT will be looked at more closely for printed circuit boards (PCB) designs utilizing boundary-scan tools in particular. One chapter is also devoted to the combination of boundary-scan with other test strategies such as In-Circuit Testers (ICT), Flying Probes and Automated Optical Inspection (AOI).

Why Test

Nobody wants to, but everybody has to. That is the simple truth for the real world in prototype and high and low volume production. No process is perfect and defects are going to happen. These defects are in some cases on a chip level with defective dies or wire-bonding issues. Other defects are more typically at the manufacturing level with incorrectly placed components, open and shorted solder joints. The manufacturing defects at the assembly (PCB) level can be detected by various testers such as Boundary-Scan testers, ICTs, Flying Probers, Manufacturing Defect Analyzers (MDAs), Functional Testers, etc. IC level defects, however, are caught by IC Testers. ICT and MDA machines require access to all the nets on the PCB, which is achieved with either the traditional Bed-of-Nail (BON) Fixture or the fixture-less testing with a Flying Probe tester.

In today's evolving technology of using smaller packages with more functionality packed into them, physical access to all the pins of the components is no longer possible in most cases. Therefore, testing would no longer be possible using the method of accessing by physical contact.

What is Boundary-Scan

Boundary-Scan or IEEE-1149.1 is a standard developed by the Joint Test Action Group (JTAG). Almost every chip manufacturer is offering many of their ICs also in an IEEE-1149.1 compliant variation. Those ICs operate in two different modes; their normal function mode and the boundary-scan mode. Identifying a boundary-scan device can be as easy as having the presence of the four (optionally five boundary-scan pins (TCK, TMS, TDI, TDO and sometimes TRST), as shown in Figure 1. However, some CPUs may have the very same pins for emulation only and are not boundary-scan compliant.

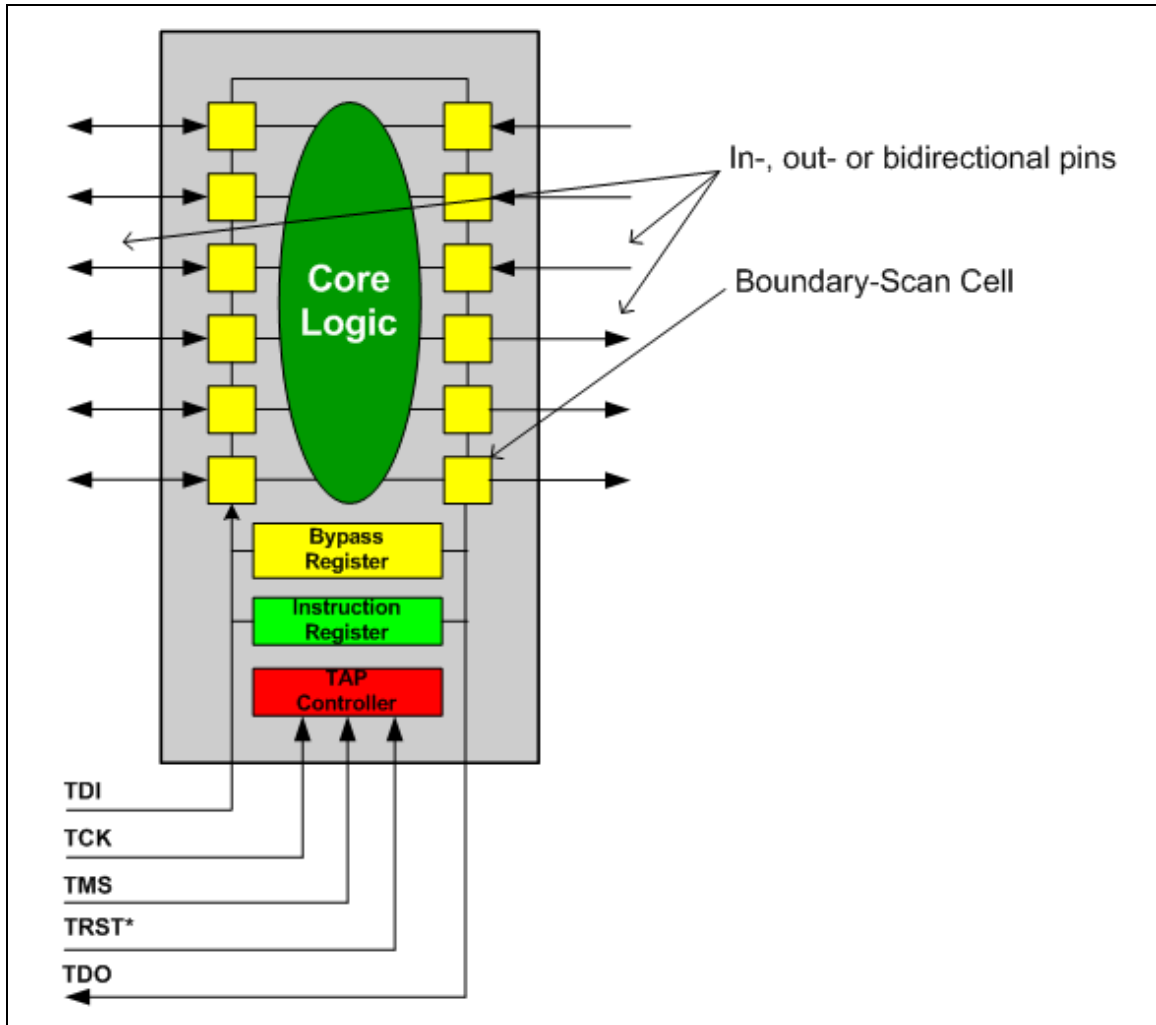


Figure 1: Block diagram of a Single Boundary-Scan Device (BSD)

When using multiple IEEE-1149.1 compliant components on the same design and interconnecting them in boundary-scan mode, they can ‘talk’ to each other with pre-defined patterns or test-vectors. Test executive software such as Corelis ScanExpress™ Runner drives this pattern. By analyzing the results, it can pinpoint any defects in regards to OPEN and SHORTED nets and indicate if a component is not functioning as it should.

An advantage of boundary-scan is that there is no need for the typical physical access to the tested nets or the components, a perfect match for today’s designs. Many designs use Ball Grid Array (BGA) packages for devices where pins are not visible and the connected net may not have any access points available at all. Testing those boards with the traditional means of using a Bed-Of-Nail tester leaves the inaccessible nets untested and any fault on it undetected.

What is JTAG Emulation Test?

If your design includes a CPU and it is supported by Corelis JTAG Emulation Test (JET), testing your design can go to the next level.

JET testing is real-time testing in ‘full-throttle’, so to speak. JET does not require any other boundary-scan devices be present in the design; the only requirement is that the processor has a JTAG port (which may be present for boundary-scan, emulation, or debug applications). With JET technology, the CPU is in complete control using the ScanExpress JET™ software, which uploads executable data to the CPU’s cache (if available) or into RAM. The CPU is then instructed to execute this code. JET has code available for automated memory testing, flash programming, and for communication to most of the peripherals attached to a CPU.

Since the code is executed by the CPU and runs at its clock and bus speed, JET allows the designer to test the board for unexpected faults which may not be detectable using regular boundary-scan tests or any other means of traditional test. Not only does JET coverage include timing and cross-talk related faults, but every bit of the memory is fully tested and in real-time. Traditional boundary-scan tests only test every *pin* of the device because testing every bit inside RAM would take too long. Test steps generated with JET are compatible with ScanExpress Runner, the test executive software for the production floor, and thus JET may be used stand-alone or in conjunction with boundary-scan testing.

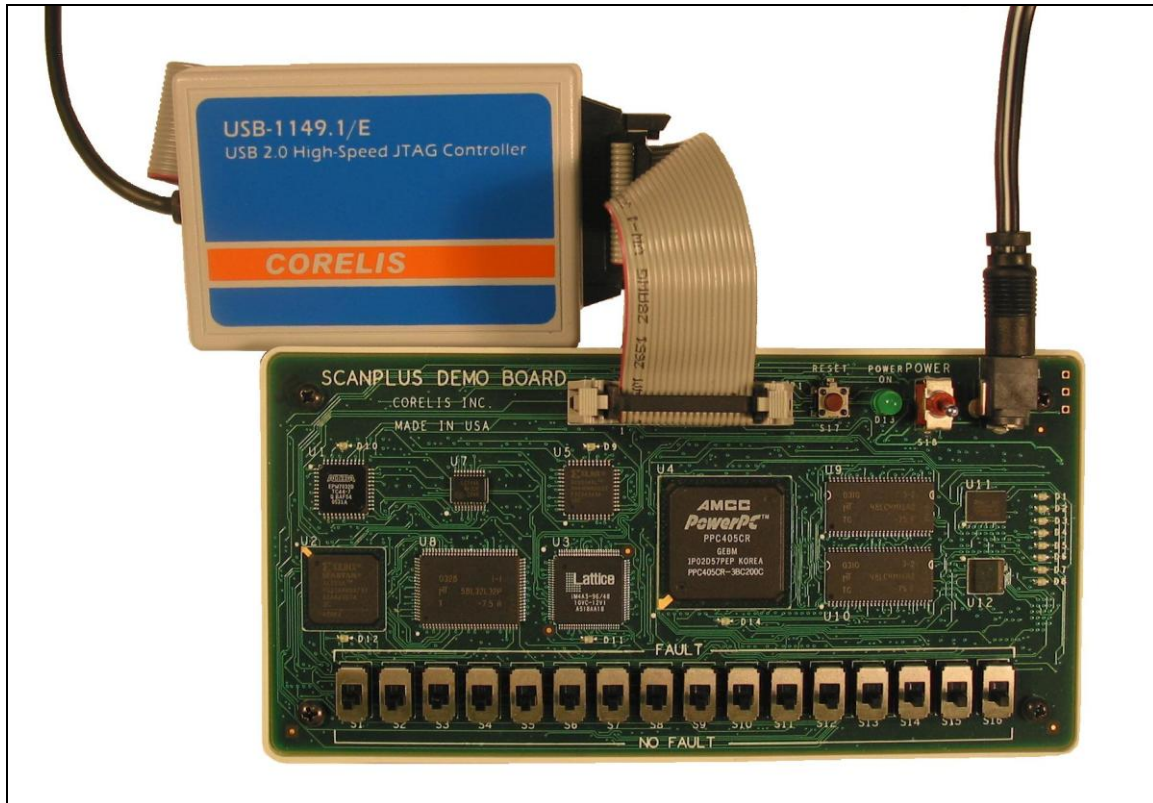


Figure 3: Target Board with Controller and Power Connected

For larger designs or designs for which remote diagnostics are desired, embedding the boundary-scan controller onto the target board with a standard connection (like USB) might be a valid option. One of those on-board controllers is the Corelis USB2JTAG device (as shown in Figure 4). The 32-pin device has one USB interface and four TAP interfaces. Instead of four headers (and the need for an external boundary-scan controller), this device connects to the signals of up to four boundary-scan chains, saving real-estate on the one hand, and by adding a standard USB connector, adding simplicity and flexibility to the design on the other.

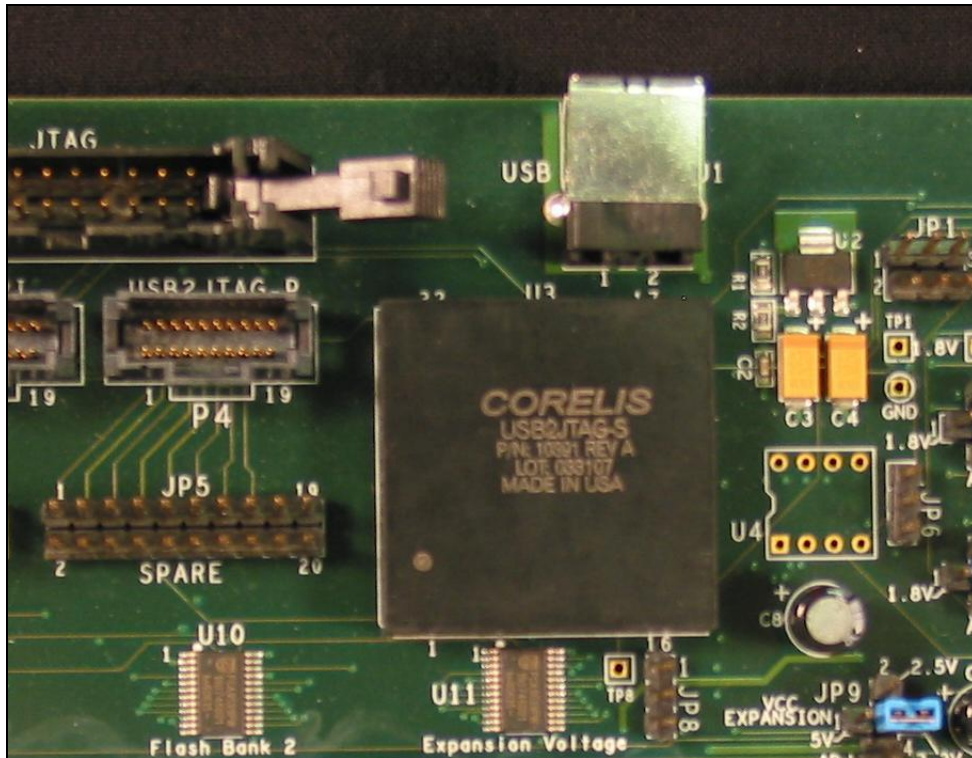


Figure 4: Incorporated USB2JTAG Controller on Board

Component Selection

Selecting the right components for your design is crucial. Not only must the choice be made between boundary-scan and non-boundary-scan devices, but also, in some cases, one has to verify that boundary-scan devices are compliant. Making the wrong choice can mean the difference between a successful boundary-scan test and no test possible.

Off-The-Shelf Devices

Here are some basic ways to find out if a device is IEEE-1149.1 compliant.

- Check to see if there are the four distinct signals on the device: TCK, TMS, TDI, and TDO.
- Find information about IEEE-1149.1 or JTAG-compliance in the data sheet.
- Ask the manufacturer of the device.

Once you have decided on a boundary-scan device, it will contain dedicated JTAG logic and a BSDL (Boundary-Scan Description Language) file provided by the manufacturer. Make sure you collect the BSDL files during this process. Typically, you can get BSDL files from the component manufacturer's web site. Corelis also maintains an internal library of BSDL files. If you have difficulty finding the BSDL file for the device that you are using, contact Corelis. The Corelis Customer Support staff may have it in their library or may be able to help you locate it.

Use Boundary-Scan-Compatible Buffers and/or Data Transceivers

This is basically a tradeoff between diagnostics capability and cost. As buffers and data transceivers increase in size (20-bit and 32-bit devices are now common), it is becoming much more cost effective to include boundary-scan capability in these devices. With Corelis ScanExpressTPG™, it is fairly easy to create cluster tests for Buffers and Data Transceivers. With cluster tests, you are basically applying patterns to one side of the buffer or data transceiver using boundary-scan devices and capturing the result using boundary-scan devices on the other side. By doing this, you are getting fault coverage over these devices, but you are not able to diagnose failures down to the pin/node level. By using boundary-scan buffers and data transceivers, you can diagnose down to the pin and node level.

Check Devices for Full IEEE-1149.1 Compliance

Before placing a component in a boundary-scan chain, verify that the component fully complies with the IEEE-1149.1 standard. Tools for Automatic Test Pattern Generation (ATPG) of boundary-scan test vectors, like ScanExpressTPG, rely heavily on the fact that the boundary-scan components are fully compliant to the IEEE-1149.1 standard. If a component is not fully compliant to IEEE-1149.1, there is a good chance that the component may not even support boundary-scan testing. In the worst case scenario, it may even be necessary to physically BYPASS the component (see *Bypassing in the Boundary-Scan Chain* section).

Devices not Fully IEEE-1149.1 Compliant

In many cases, if the selected device is not fully compliant to IEEE-1149.1, it will show the first indications in the BSDL file provided by the manufacturer. The file will have the following sections labeled as DESIGN_WARNING or COMPLIANCE_PATTERNS.

A Design Warning of the BSDL for an Altera EP20K200 looks like:

```
attribute DESIGN_WARNING of EP20K200F484 : entity is
"The APEX 20K devices support IEEE 1149.1 testing before and after "&
"device configuration; however, the devices do not support this "&
"testing during device configuration. The easiest way to avoid "&
"device configuration is to hold the nCONFIG pin low during "&
"power-up and testing.";
```

The signal `nConfig` of this device must be kept low during power-up in order to avoid configuration of this FPGA. If `nConfig` is not kept low and the device is configured, a 'Post-Config-BSDL' file will be required. Once the device is configured, the pins may no longer comply with the BSDL file provided. For more information, see the section on *Configurable BSDs (Boundary-Scan Devices)*.

A Compliance Pattern information of the BSDL for an IXP1240 looks like:

```
attribute COMPLIANCE_PATTERNS of IXP1240_BSDL_top : entity is
"(SCAN_EN, RESET_IN_L, TCK_BYN, TSTCLK, PCI_RST_L) (00000) ";
```

Design For Test (DFT) Guidelines for Boundary-Scan Testing

In order for this device to enter into boundary-scan mode, the signals listed must be placed and kept at the pattern defined prior to and during the entire test. This must be taken into consideration during the design phase of the board.

Custom Devices (ASIC)

When using Application Specific Integrated Circuits (ASICs), JTAG logic must be included in the design to utilize the device for boundary-scan testing.

Placement of the TAP Signal Pins

The TAP signal pins (TDI, TDO, TMS, TCK and TRST*) should be physically separated from one another on the component. Problems related to the TAP signals are the most difficult type of boundary-scan testing problem to diagnose. If possible, these pins should be placed adjacent to power pins on the component. The reasoning behind this is that a TAP signal shorted to a power pin is much more easily diagnosed than two TAP signals shorted together or a TAP signal shorted to another signal. With fine pitch devices, the most common type of manufacturing faults are shorted adjacent pins or open pins. Placing the TAP pins on the corners of the device reduces even further the possibility that the pin will be shorted to an adjacent pin.

Higher-Order Instruction Capture Bits

The higher-order Instruction Capture bits should be predictable. Leaving these bits as “don’t care” makes boundary-scan chain problems more difficult to diagnose. In addition, these bits can be used as a simple method of component identification when no other method exists (i.e., IDCODE is not supported).

Tri-State and Bidirectional Pin Control

Direction/high-impedance state control for certain types of bidirectional and tri-state pins (such as address and data) should be separated. If these are not separated, it may cause problems. For example, if when using boundary-scan for ISP of flash devices, the flash device itself is connected directly to the address and data pins of a microprocessor and the direction/high-impedance state of the address and data pins are controlled with the same control cell in the boundary-scan register of the microprocessor, then it is not possible to have the address pins configured as output and at the same time have the data pins configured as input, making it impossible to use the boundary-scan of the microprocessor itself to program the flash device.

Unused Instruction Opcodes

Unused TAP instruction opcodes should default to BYPASS. This ensures that the component is maintained in a safe state in the event that one of these opcodes is inadvertently selected.

Automated Tools for Design of Component JTAG Logic

When designing an ASIC, it may be worthwhile to consider using automated tools for the insertion of JTAG logic into the component. By doing this, there will be a higher probability that the JTAG logic in the component itself will fully comply with the IEEE-1149.1 standard. Boundary-scan ATPG tools, like ScanExpressTPG, rely heavily on the fact that the boundary-scan components are fully compliant with the IEEE-1149.1 standard. If a component is not fully compliant to the IEEE-1149.1 standard, there is a good chance that the component may not even be able to be used for boundary-scan testing. In the worst case scenario, it may even be necessary to physically BYPASS the component in the chain by jumpering the TDI node to the TDO node and forcing the TMS pin high. In addition, when these tools are used, a BSDL is produced automatically saving the time and effort of manually creating it.

Pin Numbers in BSDL File Should Match Those of Datasheet

In order for boundary-scan test vector ATPG tools to correlate the pin numbers correctly between the board netlist and the BSDL files for the boundary-scan components used, the component pin numbers in the BSDL files must exactly match the pin numbers in the board's netlist. Therefore, when creating a BSDL file for a boundary-scan component, the pin numbers used in the BSDL file for a boundary-scan component should exactly match those given in the component datasheet or specification.

Boundary-Scan Chain

Connectors, Signals and Termination

The IEEE-1149.1 standard specifies the signal names. However, there is no standard on the actual pin-out or mechanical definition of the connector to be used. Corelis pin configurations are commonly found in 10, 16 or 20-pin connectors. Depending on the requirements for the design, e.g., for a standard boundary-scan chain or for In-System-Programming (ISP), one of the next three pin-outs can be selected. Corelis controllers support any of the following pin-outs.

10-Pin TAP Connector

The typical schematic of a target 10-pin TAP connector and recommended termination resistors is shown in Figure 5. The 1K pull-up resistors can be connected to any V_{cc} supply with voltage between 1.25V to 3.3V, matching the boundary-scan devices in the chain. Recommended resistor values are $\pm 5\%$. Connect **all** grounds directly to the target's ground plane.

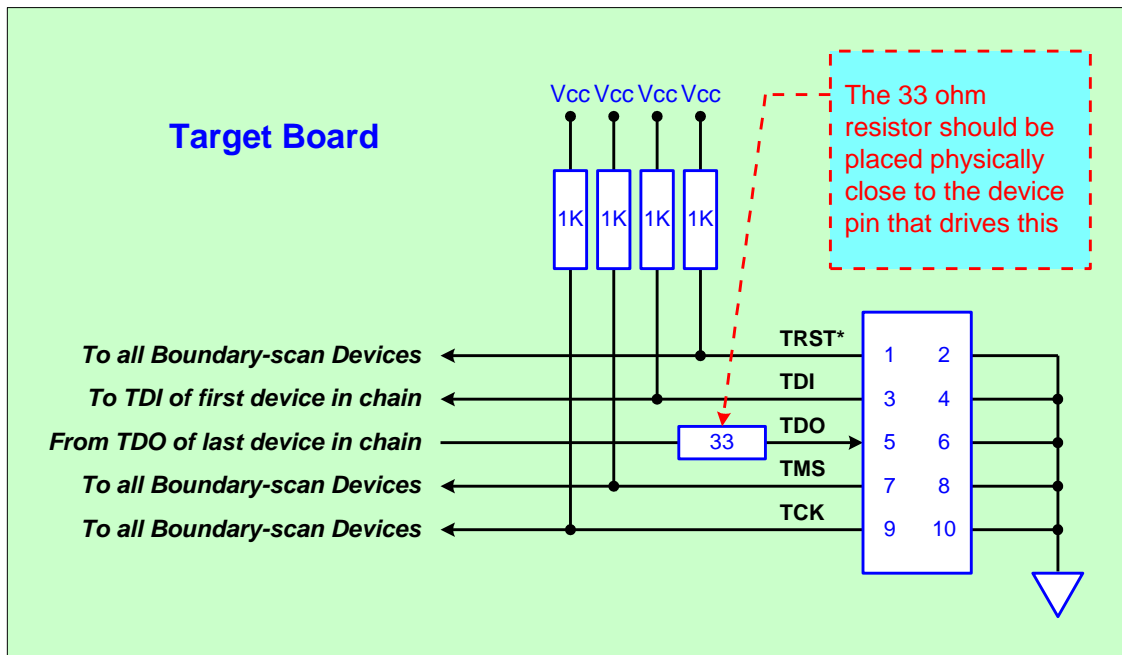


Figure 5: 10-pin TAP Connector Schematic

The 10-pin connector is mainly used for a 'straightforward' boundary-scan chain; no ISP or Direct-Write for flash is required.

16-Pin TAP Connector

The typical schematic of a target 16-pin TAP connector and recommended termination resistors is shown in Figure 6. The 1K pull-up resistors can be connected to any V_{CC} supply with nominal voltage between 1.25V to 3.3V, matching the boundary-scan devices in the chain. Recommended resistor values are $\pm 5\%$. Connect *all* grounds directly to the target's ground plane.

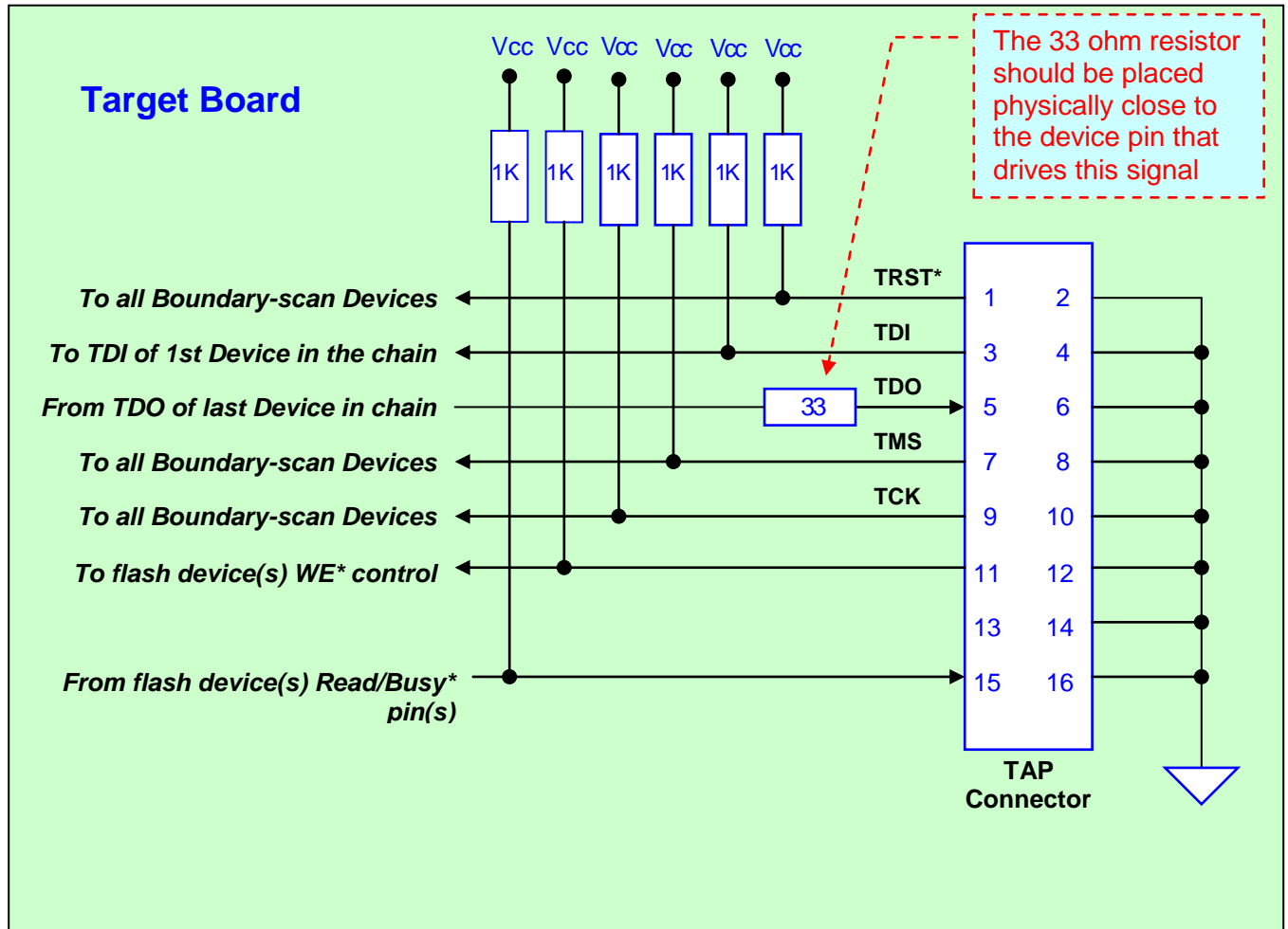


Figure 6: 16-pin TAP Connector Schematic

The 16-pin Corelis connector has three additional General Purpose Input/Output (GPIO) signals. GPIO1-3 can be used for direct flash programming or for any other general purpose needed for completing test of the Unit Under Test (UUT) or to configure the UUT for test.

Note: Some of the GPIO signals may be specifically used if JET tests are required. Contact support@corelis.com for more details.

20-Pin TAP Connector

The typical schematic of a target TAP connector and recommended termination resistors is shown in Figure 7. The 1K pull-up resistors can be connected to any Vcc supply with nominal voltage between 1.25V to 3.3V, matching the boundary-scan devices in the chain. Recommended resistor values are $\pm 5\%$. Connect *all* grounds directly to the target's ground plane.

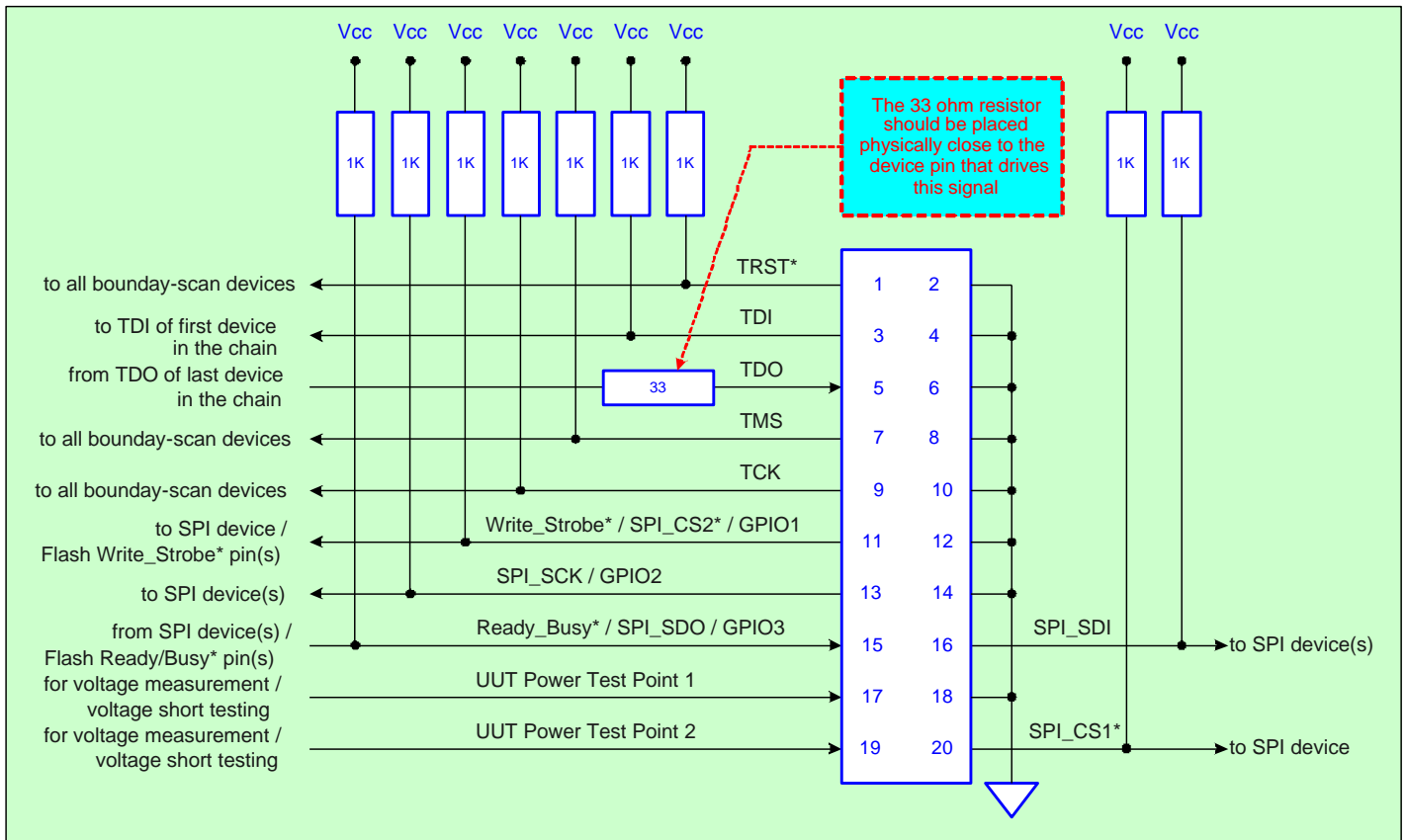


Figure 7: 20-pin TAP Connector Schematic for SPI

The 20-pin Corelis connector has three additional General Purpose Input/Output (GPIO) signals. GPIO1-3 can be used for direct flash programming or for any other general purpose needed for completing test of the UUT or to configure the UUT for test. It also supports a direct programming connection to SPI memory devices and two pins capable of voltage measurement – useful to verify the target's voltage level and check for power-ground shorts prior to testing.

Note: Not all controllers support this feature. Contact support@corelis.com for more details.

Boundary-Scan PCB-Routing Considerations

When there are a large number of boundary-scan devices on board, it may be necessary to buffer the common TAP signals, TCK, TMS and TRST*, and provide separate “copies” of these signals to individual groups of the boundary-scan components on board to reduce the loading. Doing this in no way complicates the boundary-scan testing and ISP of the board when using Corelis tools. Note that the design of components that fully comply to the IEEE-1149.1 standard ensures that a reasonable amount of clock skew between components and also between the external boundary-scan controller and the components on board (less than ¼ TCK clock period) is acceptable because:

- The state of TMS (driven by the external boundary-scan controller) changes on the falling edge of TCK.
- The component TAP state changes on the rising edge of TCK based on the current state of TMS.
- The state of TDO changes on the falling edge of TCK and the TDI value is clocked in on the rising edge of TCK.

So it is relatively simple to find a buffer that will meet these requirements.

In many cases, signal quality problems (caused by routing issues) will result in an intermittent failure of the Infrastructure test, which shifts a large number of patterns through the boundary-scan chain without affecting the I/O of the components on board. The “Multiple Run” feature of ScanExpress Runner can be used to detect and isolate signal quality problems. The degree of test failure depends on the severity of the problem. In some cases, you may only see one test failure in a thousand – but it is still detectable. It is fairly simple to set up ScanExpress Runner to run the Infrastructure test a number of times to detect these types of problems.

Group Components with Similar Logic Characteristics

In some cases, you will use devices with different logic characteristics on the same design (e.g., a mixture of 2.5V logic and 3.3V logic). By grouping the devices with similar logic characteristics together in the boundary-scan chain, it minimizes the amount of special interfacing required. However, it increases the total number of TAP connectors on your board. For simplicity, an on-board controller such as the USB2JTAG can be used to combine up to four TAPs onto one device and no external controller would be needed. See the *Real-Estate Requirements* section for more information about the USB2JTAG device.

Number of TAPs

Corelis provides single, four, eight and thirty-two TAP controllers. For many designs, a single scan chain is sufficient. It simplifies connection to the board if all of the components on board can be organized into a single boundary-scan chain. Some designs are more testable if the scan chain is split into separate chains. Any of Corelis multi-TAP controllers can combine the multiple chains into a single effective chain within the

software. If there are multiple TAP voltages, they may be grouped together by TAP voltage level. Corelis multi-TAP controllers support different voltages on each TAP for this purpose. Devices that use the JTAG port for emulation should be placed on a separate chain. This will make emulation easier to set up as some emulation solutions require a single device on the chain. Multiple TAPs will allow easier isolation of an infrastructure failure. To decrease flash programming time, the boundary-scan devices that need to toggle the flash device pins should be placed on their own chain.

Signal Buffering

Successful and reliable boundary-scan testing depends on the quality of the TCK signal. Distributing the TCK clock signal to all the boundary-scan devices on the board requires special attention. The general rule of thumb is that the TCK signal needs to be treated like any other high speed clock signal on the board. The same distribution guidelines that are applied to any other clock signal on the target board should be used in regards to loading, termination, distribution, layout-routing, matching trace impedance to termination resistor(s), keeping the clock trace as short as possible, etc. The following are some of the issues that should be considered by the circuit designer when designing for testability:

- When buffering the TCK signal, PLL-based clock buffer chips should NOT be used. Zero delay clock distribution devices (for distributing the TCK signal to multiple devices on the board) always incorporate an internal PLL and cannot be used. Unlike CPU clocks and system clocks, the TCK clock is not “free running” and cannot be used as input to zero delay clock distribution buffers as they contain an internal PLL which requires a free running input clock source. For example, the Cypress CY2309 or the Pericom PI6CV857L are clock driver chips that incorporate a non-PLL bypass mode of straight-through clock-in to clock-out mode (for testing).
- Series resistors on clock driver’s outputs should be included in the design.
- While a single TAP makes it easy to connect and test, the user can implement multiple TAPs to connect to different sections of the target board. It is generally recommended to restrict the total number of TAPs to four or less.
- The size of the board and the number of boundary-scan-compatible components may require separate TCK signal buffers for each group of physically adjacent boundary-scan devices.
- The TDO, TDI, TMS and TRST* signals are less sensitive than TCK. The 1K ohm pull-ups are generally sufficient as termination and typically no other special considerations are required.
- Using an active cable could help improve the TCK signal quality. The active cable places the buffer closer to the tip of the cable that plugs into the customer target board.
- Splitting the board into multiple TAPs helps segment the board into small adjacent boundary-scannable clusters. However, it requires extra connectors and a Corelis multi-TAP controller.

- In a chassis with multiple cards the boundary-scan can be configured in a multi-drop configuration where a single TAP is bused across the backplane. A multidrop scanchip such as the National Semiconductor Scanbridge can be used to buffer all TAP signals from the host and distribute them reliably to multiple local TAPs.

By-passing

In some cases (such as for JTAG emulation), it is required to keep certain boundary-scan components (such as a microprocessor) isolated in a chain by themselves. This can be done using jumpers or through the addition of electronic components to the board design. Shown in Figure 8 is an example of using jumpers to make the boundary-scan chain reconfigurable.

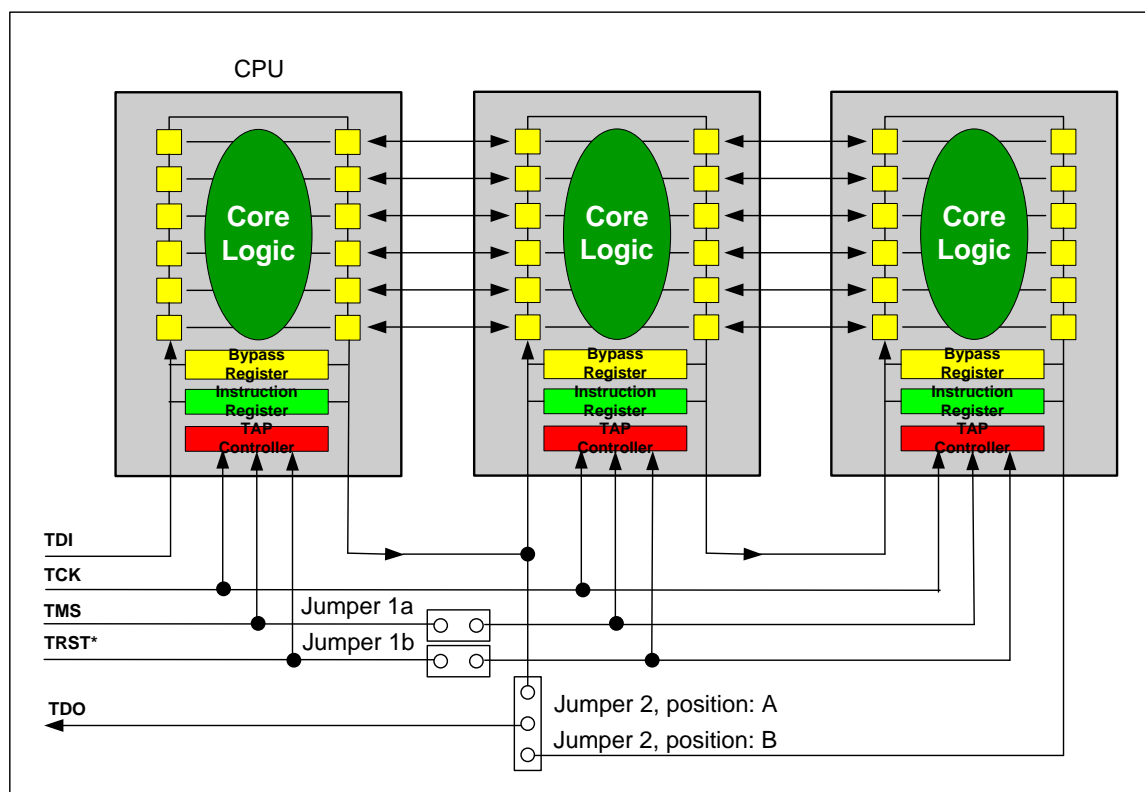


Figure 8: Bypassing manually with jumpers

In Figure 8, two jumper locations are used to determine whether just the microprocessor is in the chain or all JTAG devices are in the chain. By installing Jumper 1 (a + b) and also installing Jumper 2 on position B, all devices are included in the chain. If Jumpers 1 are removed, the internal pull-up resistors will hold the TMS and TRST* on the "other" JTAG devices in a high state, keeping them in Test-Logic-Reset. If a jumper is then installed on Jumper 2 position A, the TDO from the microprocessor is routed to the connector. This effectively places only the microprocessor in the chain.

Figure 9 shows an example of including additional logic devices in the boundary-scan chain in order to make it reconfigurable automatically:

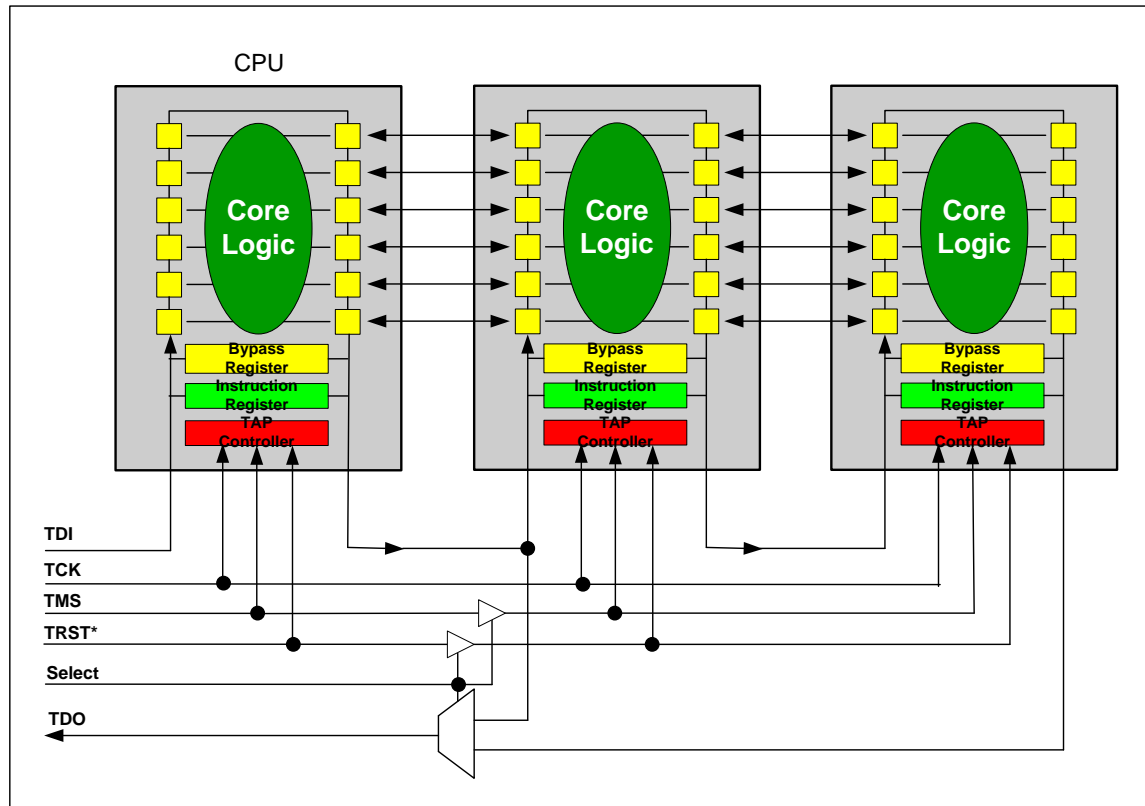


Figure 9: Bypassing with added logic

Figure 9 is a similar arrangement to Figure 8, but uses logic (multiplexer buffers with OE) instead of jumpers. A "select" at the connector is used to determine the chain configuration. Grounding the "select" will just include the microprocessor in the chain. If the "select" is logic "1", then all devices are included. An advantage of this arrangement is that it allows controlling the JTAG chain configuration based on the cable that is plugged into the JTAG connector. The 'SELECT' signal could be one of the GPIO pins available on the Corelis 16- or 20-pin TAP connectors.

Avoiding Problems with the TAP Signals

To avoid problems with the scan chain, do not connect I/O pins to the TAP signals. It is likely that the infrastructure test will not work correctly and the board may have to be redesigned. Figure 10 and Figure 11 show some common connections to watch out for.

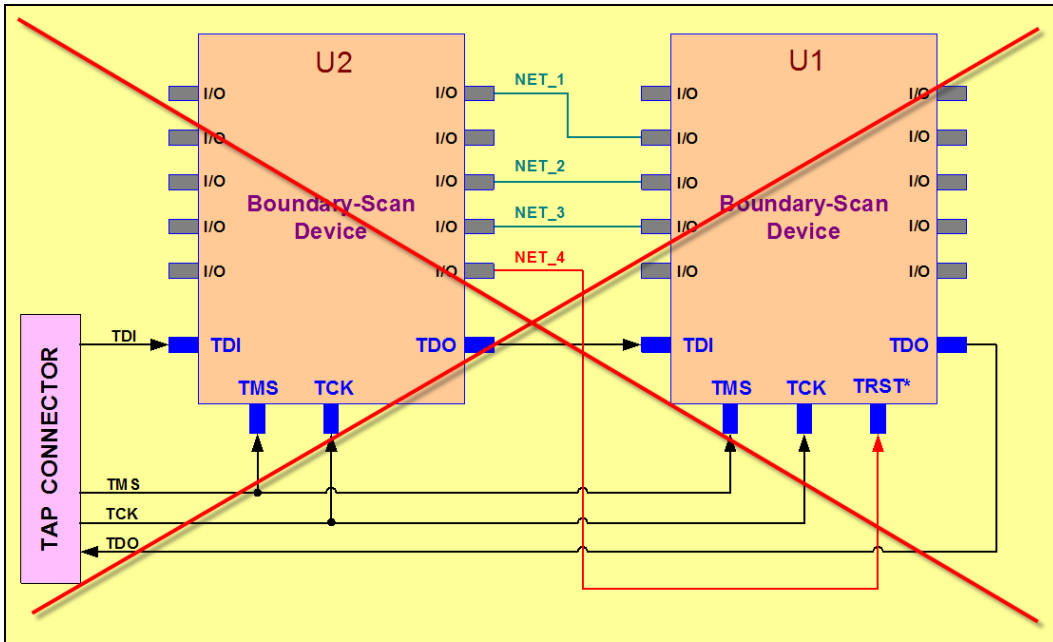


Figure 10: Do Not Connect TRST* to I/O Pins of Devices in the Scan Chain

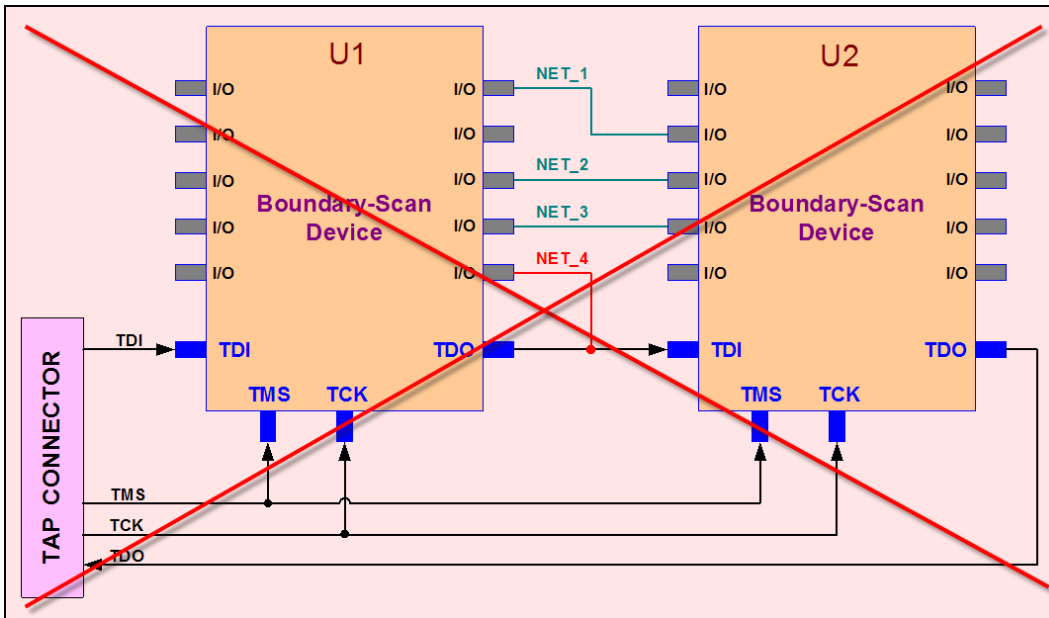


Figure 11: Do Not Connect the TAP Signals to I/O Pins of Devices in the Scan Chain

Configurable BSDs (Boundary-Scan Devices)

Most BSDs are brought into boundary-scan mode by either using the TRST* signal or by using a combination of TCK and TMS signal levels. However, some BSDs require additional signals to be set to a defined level in order to be put into boundary-scan mode.

Other BSDs (specifically FPGAs) can be configured during power-up and, as a result, behave differently in boundary-scan mode.

The following are important considerations when using configurable BSDs.

Compliance Enable Pins

Compliance enable pins are pins that need to be controlled in a particular way to allow the device to operate in boundary-scan mode. Various aspects of compliance enables pin on boundary-scan devices are discussed in this section.

The section “*Testing with Programmed and Configured Devices*” below also contains information that is relevant to compliance enables that may extend beyond the information in this section. The most important recommendation is to avoid configuration for reliable testing.

Configuration pins

Configuration is prevented by pins, some of which may be listed as compliance enable pins in the BSDL file, and some are found in the data sheet. Examples: Altera nCONFIG; Xilinx INIT and PROG (also named PROG_B, PROGRAM in some data sheets).

Boundary-scan while configuring

Most devices do not support boundary-scan testing while configuring. If the board testing starts immediately after power up, boundary-scan testing may start while on-board configuration is occurring – these tests typically fail. If the test is re-run, it may pass. This will appear to be an intermittent failure to the tester. It will require that the tester either delay the start of test after power up, or observe the “programming complete” LED (if there is one) before starting testing.

Post-configuration BSDL

After configuration, the device supports boundary-scan. The complication is that the BSDL file provided by the manufacturer is for an un-configured device. It may not match the boundary-scan behavior of a configured device. A “post-configuration” BSDL file may be required. The device manufacturer provides software tools to create the post-configuration BSDL file.

Boundary-scan compliance vs. “normal” mode

In many designs and devices, the settings for compliance enable pins in the “normal” operating mode is not compatible with the settings for boundary-scan testing mode. Some situations decrease test coverage; some situations completely disable the scan

chain, and thus disable boundary-scan testing. To support both situations, some board designs require modifications.

Avoid compliance “workarounds”

There are many situations that have a workaround for a design or device compliance enable “problem”. We recommend against relying on workarounds because this adds risk. Using a workaround increases the risk that something will cause the workaround to stop working. Some of the workarounds involve test execution options that must be set, which increases the chances that the test will stop working.

BSDL “COMPLIANCE_PATTERNS”

Compliance enables are listed in the “COMPLIANCE_PATTERNS” section of the BSDL file. The syntax is a list of pin names, followed by a list of ones and zeros that indicate the level the pins must be at for boundary-scan operation.

Non-compliant “COMPLIANCE_PATTERNS”

There are pins that are actually compliance enable pins from a testing point of view, but do not strictly meet the IEEE-1149.1 definition. These are not listed in the “COMPLIANCE_PATTERNS” section of the BSDL file. Instead, they are listed in the “DESIGN_WARNING” section or in comments or not listed at all. They can sometimes be found in data sheets, application notes and errata for the device. “JTAG” and “scan” are good search phrases.

Example: Some CPUs require loading in a configuration on power up. The configuration selects the use of the JTAG port pins as either for emulation or for boundary-scan testing. This is not listed as a compliance condition anywhere in the BSDL file, but it obviously affects whether or not the device will support boundary-scan testing.

Different types of compliance enables

There are several “types” of compliance enables. They include any combination of the following characteristics:

- Needs to be at the correct state from power up and continuously all through testing.
- Needs to be at the correct state only during testing.
- Has an internal pull-up/pull-down to set it to the correct state.
- If not compliant, the device does not support boundary-scan operation except while the device is configuring.
- Must be set on power up; afterwards it doesn't matter.
- Requires a minimum number of clocks at the specified level after power up.

Compliance pin effects

Compliance pins may have a number of effects, including:

- Some may only enable some of the I/O pins and other I/O pins are not affected.
- Some prevent the device from powering down.
- Some cause the JTAG state machine to stop working, causing a break in the scan chain.
- Some prevent the device from configuring.

Device specifications

The manufacturer may not specify all of the above factors in any of their documentation.

Controlling compliance pins

There are a variety of ways as to how the control of compliance enable pins is implemented in designs:

- Pulled high/low correctly with a resistor.
- Pulled high/low incorrectly with a resistor.
- Is connected to another boundary scan-device I/O pin.
- Is connected to a programmed CPLD that will set it incorrectly when the CPLD is in non-boundary scan mode (bypass or between test steps), but can set it correctly during boundary-scan mode.
- Is floating, but has an internal pull-up/pull-down to set it to the correct state.
- Is floating, but has an internal pull-up/pull-down to set it to the correct state. The internal pull-up/pull-down is internally disconnected when the device is in HIGHZ.

Incomplete device documentation

Unfortunately, the BSDL files and device documentation are not always complete. Sometimes what is really needed is discovered only during the test procedure integration with the target.

Always observe compliance rules

There are some designs that do not fully comply with our recommendations and still test fine. The problem is that it can be difficult to know all the design and device details in order to reliably predict the outcome. To minimize the risk of finding out too late (i.e., after the PCB is laid out and fabricated), we recommend meeting the compliance enables from power up and all through testing in the design. Sometimes a seemingly minor change in parts, PCB, CPLD programming, etc., will suddenly cause boundary-scan to stop working even though it worked before.

Devices may exit boundary-scan model unexpectedly

Note that when ScanExpress Runner runs the test steps, devices can go out of boundary-scan mode between the test steps.

Example: If a programmed CPLD is controlling the Altera FPGA `nCONFIG` compliance enable pin, and the CPLD is programmed to drive it high, the Altera FPGA may run the first test step fine, but then gets incorrectly driven between the test steps, allowing it to go out of boundary-scan mode and into programming mode between the 1st and 2nd step. Then the second step doesn't work because the Altera FPGA is no longer in boundary-scan mode.

Specifying “constraints” is not sufficient

Do not connect compliance enable pins to other devices in the scan chain. Constraints in the ScanExpress constraint (.CON) file should not be relied upon to control compliance enable pins. There is a common misconception that if a compliance enable pin needs to be high, using `FIXED_HIGH` in the constraints file is adequate to allow proper boundary-scan operation. All the devices in the scan chain must be in boundary-scan mode to allow controlling pins in boundary-scan. If some of the compliance enable pins are not set correctly by hardware, one or more devices may not support boundary-scan and the scan chains may not operate properly. Note that at power up, the board is operating “normally”, and is therefore not under boundary-scan control. The normal operation of the board may not set the compliance pins to the correct level.

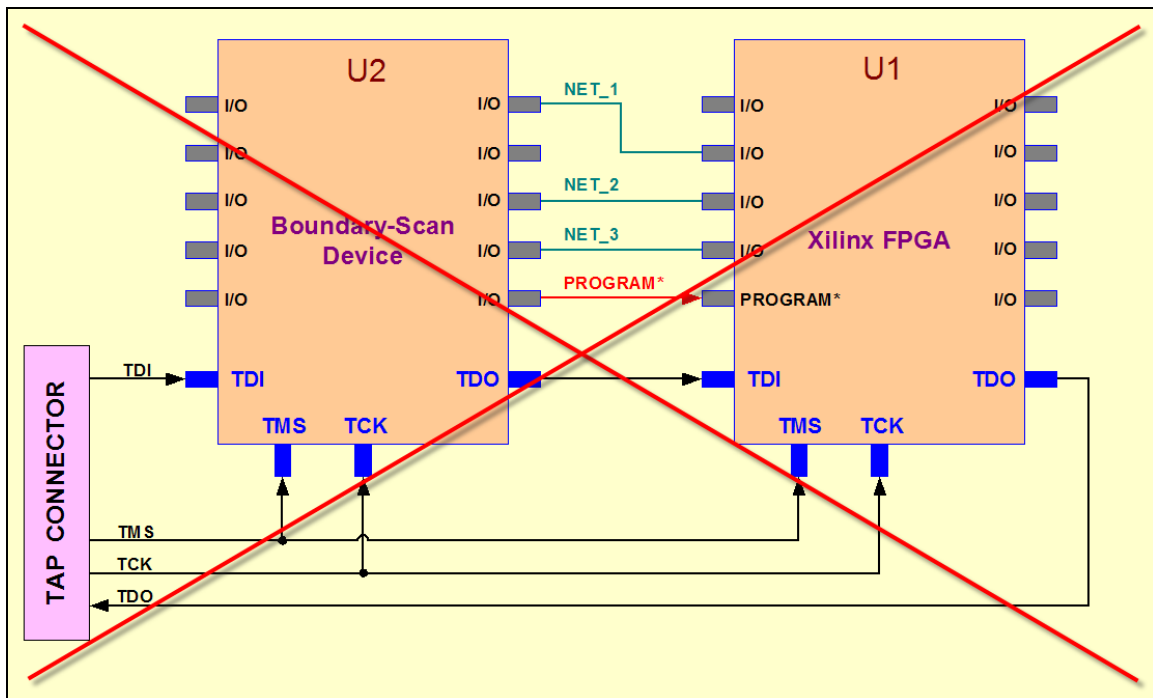


Figure 12: Do Not Connect Compliance Enable Pins to Other Devices in the Chain

Using jumpers or test points to force compliance

Compliance enable pins can be forced to the correct level for boundary-scan testing using jumpers, shunts, switches or fixture test points. The shunt is installed during board test and open for normal operation. Similarly, the fixture test point forces the compliance enable net to the appropriate value when the board is installed in the fixture for testing. Care should be taken during the design of these circuits to avoid damaging any other devices that may be actively driving the net.

Figure 13 shows another useful technique that can be applied to open collector compliance enable signals. This method is frequently used to prevent FPGAs from configuring during boundary-scan testing. The Altera nConfig or Xilinx INIT_B signals are pulled up and brought to pin 4 of the Corelis TAP connector. The pin is grounded by the boundary-scan controller when it is connected for testing and the FPGA is prevented from configuring. When the boundary-scan controller is removed, the signal reverts to its original state allowing the FPGA to configure and the board to operate normally.

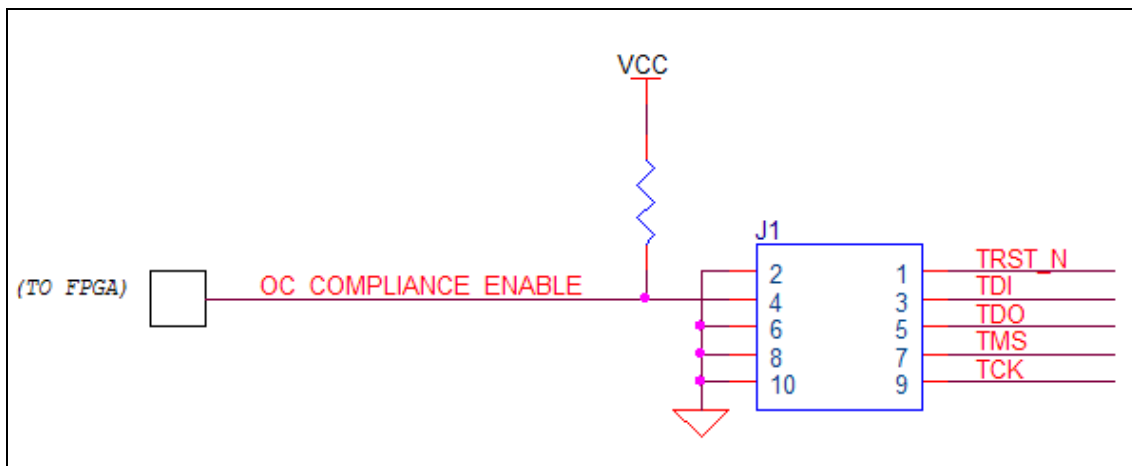


Figure 13: Using the TAP Connector to Force a Compliance Enable Signal Low

Power pins as compliance pins

In effect, the device power pins are compliance enable pins.

Example: There is a net that turns on/off the power supply on the board. If this net is not at the correct level, power to the boundary-scan device is not provided. The scan chain will probably fail. Therefore, the control net must be at the correct level from power up and throughout testing.

Additional precautions

While the above rules are very simple, we recognize that implementing these rules and meeting design requirements can be involved.

- In the design, provide a hardware method that can control the compliance enable pins to be at the correct level from power up and continuously all through boundary-scan testing. The level must be guaranteed to be stable.

Note 1: As a corollary, using programmable devices to control compliance enables is not recommended unless it is implemented very carefully to support both normal operation and boundary-scan operation, whether the device is programmed or not.

Note 2: A floating pin is not guaranteed, even if there is an appropriate internal pull-up/pull-down resistor. We recommend using an external resistor to control the compliance enable and not rely on the internal resistor.

Note 3: Because of the vagaries of compliance enables, a board can be testable even if the compliance enables are not met. We recommend not assuming that this will always work. Basing other designs on this “working” design is a risky design decision. It often happens that the first design experiences no problems because all the variables are “correct”. But since all the variables are difficult to keep track of (if they are even known), another design may not set all the variables correctly for boundary-scan.

- For any compliance enable pins that can be driven by boundary-scan, constrain the net to the correct level in the constraint file.

Note 1: This is done to prevent boundary-scan controllable pins from toggling the compliance enable pins. This can cause contention or change the compliance level.

Note 2: Doing this does not satisfy the requirement that the compliance enable pin be set at the correct level from power up and all through testing.

Testing with Programmed and Configured Devices

There are many effects when testing with programmed and configured devices which may compromise boundary-scan testing. In general, we recommend testing with un-configured devices. Below are some situations where configuration is required. Also be sure to observe the applicable rules concerning compliance pins described above in *Compliance Enable Pins*.

BSDL files for pre-configuration and post-configuration

The BSDL file, as distributed by the manufacturers, is for un-configured devices. As many recent BSDL files warn, the BSDL file may need to be altered to match the post-configuration operation of the device. If the post and pre-configuration BSDLs are different, the test generation and test procedure are more complicated. If a pre-configuration BSDL is used for a post-configured device, predictable operation is compromised. The following are some factors.

- If testing with a configured device, some device manufacturer tools have the ability to generate a BSDL file based on the configuration.
- The configured BSDL file may change whenever the configuration changes. This means that the tests may have to be re-generated every time the BSDL file changes.
- This adds to the logistics due to keeping track of the different configurations and the tests appropriate for that configuration.

CPLD-like devices

Some devices act like CPLDs, but are not true CPLDs. They internally “configure” on power up, loading data from an internal flash to internal SRAM. They may therefore require a post-configuration BSDL file. During internal configuration, just like FPGAs, boundary-scan testing is not supported. These devices include the Altera MAX II and Xilinx CoolRunner families.

Devices that do not support the EXTEST instruction during configuration

Devices may not support the EXTEST boundary-scan instruction when they are configuring. They only support EXTEST pre-configuration and post-configuration. See their application notes. If for some reason the FPGAs start configuration but do not finish, then EXTEST is unavailable. The chances of this occurring are very small, but we have seen this occur.

Long configuration times

In some designs, the time it takes for all the devices to complete configuration is significant. This then requires a delay after power up before testing can begin. If the testing is automated, testing may start very soon after power up, resulting in failures.

Tri-stated pins when in BYPASS

Before configuration, most devices tri-state their pins when the device is in BYPASS mode. Pins set to the safe mode are tri-stated. After configuration, the pins may go to high or low during BYPASS and when set to their safe value. This makes the device and board appear to act less predictably.

JTAG pins configured as I/Os

Some devices allow the JTAG pins to be changed to I/O pins by the programming. This will not allow boundary-scan to operate after configuration. Most modern devices do not allow this. Keeping the JTAG pins as JTAG pins may be a compilation option, and the default may be to use them as I/O.

Safe mode between test steps

After the end of each ScanExpress Runner test step, the pins are placed in their safe value. Before each test step, by default, all devices are reset (by TMS), which places the device in BYPASS mode. This is least harmful for the board since it reduces contention between devices and allows for consistent testing. If the device is configured, its behavior may vary with the programming, adding to the variables in testing.

Design to prevent configuration

While some boards will test successfully after configuration, if it later turns out to be a problem, it is better to already have the ability to prevent configuration. We therefore recommend incorporating into the design the ability to prevent device configuration when the device has pins to do that.

Test with pre-configured CPLDs and FPGAs

CPLDs and FPGAs are almost ideal for boundary-scan testing. Each pin supports input and output simultaneously and each pin has its own output control cell. Any post-configuration BSDL is therefore less than ideal, possibly resulting in reduced coverage.

Testing with configured devices

In some cases, the board must be tested with configured devices due to the board assembly process. For boundary-scan testing, usually a device can be erased, the board tested, then the device re-programmed. There are cases, however, when this method is not viable.

Disadvantages of testing with configured devices

In some cases, configuration is required due to the board design. Whenever possible, we recommend against designing a board this way. The board may work functionally, but this can have negative effects on board testing. Testing with configured devices may:

- Add requirements or conditions to board testing.
- In extreme cases, the board may not be boundary-scan testable.

Design For Test (DFT) Guidelines for Boundary-Scan Testing

- Decrease test coverage.
- Increase board rejects when the board failure mode prevents the workaround from testing.
- Increase the difficulty to debug.
- Require workarounds.

A workaround usually requires that certain conditions be met. The next board revision may not meet all the conditions or may alter the conditions and thus the workaround may not be viable for the new board revision.

Sometimes these types of boards need to be tested because they are already built. Some situations require board rework. Some situations have boundary-scan workarounds. For example:

- The **device enables power to a section of the board**, and it does this only after configuration. Without configuration, the section remains unpowered. This section will not support boundary-scan until it is powered up.
- A board has **two power sections** and both sections have boundary-scan. Section 1 powers up when the board is powered up. Section 2 powers up based on the outputs of a section 1 programmable device. This device must be configured to correctly power up section 2.
- The **device controls compliance enable pins**. Un-configured, the compliance enable pins will not be set to support boundary-scan.
 - Case 1: Configured, the compliance enable pins will be set to support boundary-scan.
 - Case 2: The functional configuration also does NOT set the compliance enable pins to support boundary-scan.
- The **device controls MUXes** in the scan chain. The desired topology is attained only after configuration.
- The device **default I/O voltages are not appropriate** for a device on the bus. Only after configuration are the voltages correct. This situation can occur with low voltage memories or differential interfaces.
- A board **watchdog cannot be disabled**. When it triggers, it interferes with boundary-scan. Boundary-scan is too slow to provide the watchdog “keep alive” input. A configurable device must be programmed to provide the watchdog’s “keep alive” signal.
- Some situations require **higher drive than the un-configured drive** characteristics, e.g., DDR2 memories. These memory devices frequently have termination resistors on the DDR2 signal lines, usually around 51 Ohm to a termination voltage, equal to one half of the DDR2 operating voltage. This termination requires strong drivers.

Design For Test (DFT) Guidelines for Boundary-Scan Testing

Below are some resolutions and workarounds. Any method used should avoid damage to the board. For example, installing a shunt for testing should not cause contention.

- Use a **test-only configuration** that sets the outputs to the required levels. All other I/O should result in bidirectional boundary-scan cells, with simultaneous drive and pin read-back, and with output control on all pins. This maximizes test coverage and diagnostics.
- Use the **production device configurations**. This requires that testing accommodate that the device is programmed. It may require input pins to be at certain levels. It may have outputs whose nets should not be driven during boundary-scan testing. Below are the cases and the recommended actions:

Case	Action
Always high output	Sense high during boundary-scan testing
Always low output	Sense low during boundary-scan testing
Output that will not be a known steady value	Exclude from boundary-scan testing
Input that should be driven high if a boundary-scan device is connected to it	Fix high during boundary-scan testing
Input that should be driven low if a boundary-scan device is connected to it	Fix low during boundary-scan testing

- **Add a test mode** to the device configuration. Use an input pin to the device to switch between production mode and test mode. Design the test mode to meet boundary-scan testing requirements. Provide control of the mode pin in the board design.

General Configuration for Xilinx INIT and Altera nCONFIG

The Xilinx INIT and Altera nCONFIG FPGA pins prevent the device from configuring. During configuration, the FPGA is not available for boundary-scan operations. Post-configuration, the I/O pins will be configured according to the programming. Here are some considerations regarding these pins.

Design the pins to be low for boundary-scan testing

In general, as a design practice, we recommend that these pins be low for boundary-scan testing, from power up and all through testing. This will allow for more reliable testing.

Testing when pins are high

Note that in many designs, for various reasons, testing will run successfully even with these pins high. This depends on the specific FPGA, whether there is a configuration

device (such as a serial EEPROM), the timing of when testing starts, the programming of the I/O pins after configuration, how the boundary-scan devices are connected to each other, and possibly other device-dependent factors.

BSDL file considerations

The BSDL file, as distributed by the manufacturers, is for un-configured devices. As many recent BSDL files warn, the BSDL file may need to be altered to match the post-configuration operation of the device. If the post- and pre-configuration BSDLs are different, the test generation and test procedure are more complicated. If a pre-configuration BSDL is used for a post-configured device, predictable operation is compromised. Some consequences of needing to use a post-configured BSDL file are:

- If testing with the device configured, sometimes the device manufacturer has the ability to generate a BSDL file based on the configuration.
- The configured BSDL file may change whenever the configuration changes. This means that the tests may have to be re-generated every time the BSDL file changes.
- This adds to the logistics due to keeping track of the different configurations and the tests appropriate for that configuration.

Configuration and EXTEST

The FPGAs do not support EXTEST if they are in the process of configuration. They only have EXTEST in pre-configuration and post-configuration. (See their application notes). Therefore, if for some reason, the FPGAs start configuration but do not finish, then EXTEST is unavailable. The chances of this occurring on any particular board are very small, given the total nets versus the number that are involved in configuration process, but we have seen this occur.

Long configuration times

In some designs, the time it takes for all the FPGAs to complete configuration is significant. Keeping the FPGAs un-configured allows boundary-scan to be started immediately after power-up. This helps where testing time is especially critical, or the testing automatically starts immediately after power up.

Tri-stated pins when in BYPASS

Before configuration, most FPGAs tri-state their pins when the device is in BYPASS. After configuration, the pins may go to high or low during BYPASS and when set to their safe value. This makes the device and board appear to act less predictably. This can result in zero, a few, or a large numbers of failures. If there are a large number of failures, it is likely that the programmed device is directly or indirectly causing scan chain failure.

JTAG pins configured as I/Os

Some FPGAs allow the JTAG pins to be changed to I/O pins by the programming. This will not allow boundary-scan to operate after configuration.

Safe mode between test steps

After the end of each ScanExpress Runner test step, the pins are placed in their safe value. Before each test step, all devices are reset (by TMS), which places the device in BYPASS mode. This is usually least harmful for the board since it reduces contention between devices and allows for consistent testing. If the device is configured, its behavior may vary with the programming, adding to the variables in testing.

Specifying “constraints” is not sufficient

Constraints in the constraint (.CON) file should not be relied upon to control these pins. There is a common misconception that using `FIXED_LOW` in the constraints file to control a compliance enable pin that needs to be high is adequate to allow proper boundary-scan operation. All the devices in the scan chain must be in boundary-scan mode to allow standard boundary-scan control. If all the compliance enable pins are not set correctly by hardware, one or more devices may not support boundary-scan. They therefore cannot control pins connected to compliance enables. Note that at power up, the board is operating “normally” and is therefore not under boundary-scan control. The normal operation of the board may not set the compliance pins to the correct level.

Recommendations:

- In the design, provide hardware of control compliance enable pins at the correct level from power up and continuously all through boundary-scan testing. The level must be stable. For that reason, using programmable devices to control compliance enables is not recommended unless it is implemented very carefully to support both normal operation and boundary-scan operation, and whether the device is programmed or not.
- For any compliance enable pins that can be driven by boundary-scan, constrain the net to the correct level in the constraint file. This is done to prevent contention with the hardware control of the compliance enable pin. Doing this does not satisfy the requirement that the compliance enable pin be set at the correct level from power up.

Design to set `nCONFIG` and `INIT` pins low

While some boards will test successfully without the `nCONFIG` and the `INIT` pins low, if it is a problem, it is better to already have the ability to set them low for boundary-scan testing designed into the board.

Alternatives to grounding `INIT` or `nCONFIG`

The following are some testing approaches if Xilinx `INIT` or Altera `nCONFIG` is not able to be grounded in the design.

- Case 1: Short `INIT` to ground for boundary-scan testing. This is the best for test coverage, but may be impractical for the test environment. In some cases, if a fixture is used, and there is a test point on the net, the fixture can ground the net.

Note that for most SEEPROMS that drive this net, its output is open collector, so shorting the net to ground will not damage the SEEPROM.

- Case 2: Do not short INIT to ground for boundary-scan testing. Use a post-configuration BSDL file for testing. See discussion above.
- Case 3: Do not short INIT to ground for boundary-scan testing. Exclude the nets that now fail from testing. Note that the nets that fail may change depending on the device configuration.
- Case 4: Do not short INIT to ground for boundary-scan testing. Erase the configuration SEEPROM after infrastructure testing, but before all other testing. Cycle power on the board. This will result in an un-configured FPGA in most cases.

Control of Non-Boundary-Scan Devices

Some non-boundary-scan devices include the ability to place all of the outputs in a high impedance state (tri-state). In some cases, this is done when the reset signal of the device is asserted to the device. In other cases, it is a dedicated pin such as the OE or CS pin or the ONCE pin on some Motorola devices. Asserting these pins can be done by using a dedicated jumper which is installed on-board during Boundary-Scan testing. It can also be done using the Parallel I/O (host bus controlled I/O) available on the Corelis boundary-scan controllers.

For state machines and other types of synchronous devices (such as synchronous memory SDRAM and SSRAM), it is critical to have control of the clock to the device with a boundary-scan controllable pin. If caught early in the design stage, there are methods of providing control of the clock signal with boundary-scan and not affecting the overall operation of the circuit. For example, if a tri-state-capable clock driver/generator is used in the design and if it is possible to tri-state this for purposes of testing (using a jumper or other mechanism), then a normally "unused" boundary-scan pin (from an FPGA, for example) can be attached to this net. This boundary-scan pin will only be used for testing and will not be used during the normal operation of the circuit. This will give you boundary-scan control of the net when testing, yet not affect the operation of the circuit functionally.

Testing Memory Devices using Memory Clusters

In today's market, the majority of memory devices available do not include boundary-scan capability. Even if the on-board memory devices do not include boundary-scan capability, they can still be tested automatically using ScanExpressTPG through what is referred to as a **Memory Cluster test**. Memory Cluster tests are produced automatically by ScanExpressTPG for many types of memory devices, including Async DRAM, Sync DRAM, Async SRAM, Sync SRAM, Async FIFOs, Sync FIFOs, Registers, etc.

Memory cluster tests are performed by emulating read/write cycles to the memory device using surrounding boundary-scan devices. The patterns in this case are applied to the memory device not in real-time, but at a much slower speed. However, memory cluster

Design For Test (DFT) Guidelines for Boundary-Scan Testing

testing will give you the capability of detecting opens and shorts on the pins of the memory device, which are the most common types of problems found in manufacturing and production. Therefore, when testing memory devices that do not include boundary-scan capability, it is critical to have the ability to control the pins on the memory device using surrounding boundary-scan devices. This includes, but is not limited to, address pins, data pins, chip select/enable, output enable, RAS (DRAM), CAS (DRAM), clock (synchronous devices), write strobe (FIFO), read strobe (FIFO), and write enable signals.

Note that in addition to this, the boundary-scan device(s) that are used to apply patterns to the memory device must have the capability of driving the address and control pins on the memory device, and independently have the ability to drive or sample from the data pins on the memory device. On some boundary-scan devices, multiple outputs on the device are assigned to the same control cell in the boundary-scan register, limiting the way the outputs are configured/controlled.

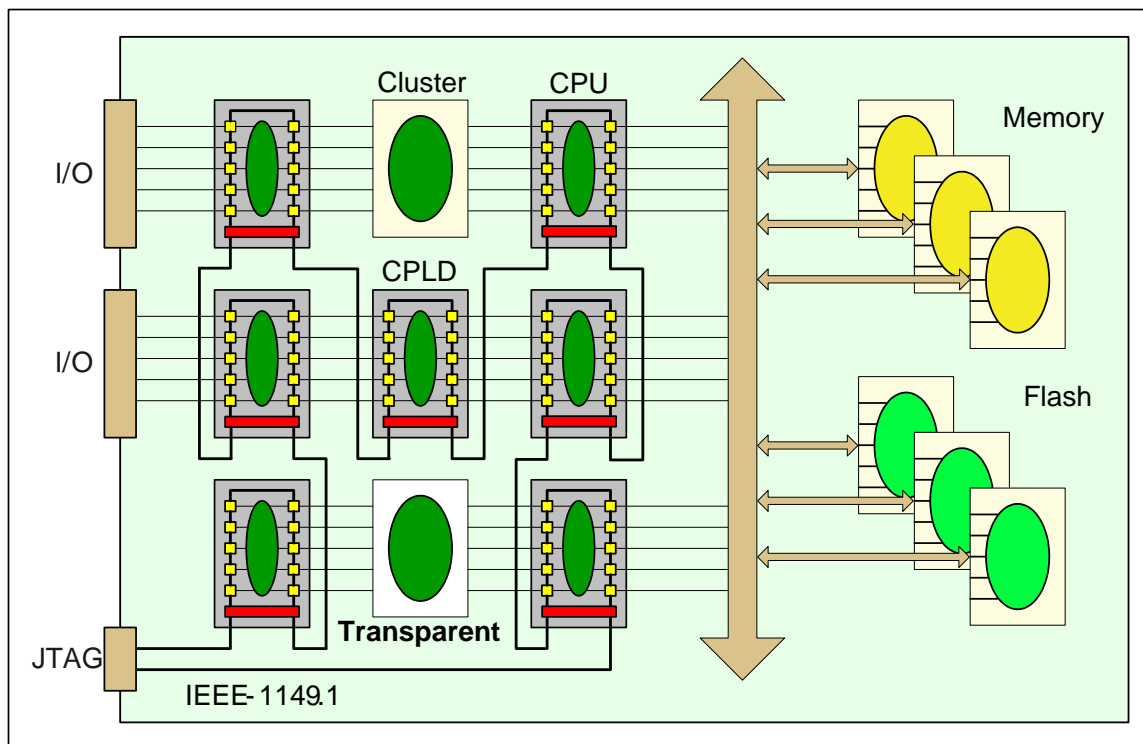


Figure 14: Memory Devices Surrounded by Boundary-Scan Devices

Clock-Driven Memory SDRAM DDR

In order to test SDRAM and DDR memory devices as boundary-scan clusters, it is required that the entire SDRAM chip be "surrounded" by boundary-scan devices, including the clock signal. Often, the clock signal of the SDRAM device is driven directly by a special clock driver/PLL chip which is not boundary-scan compatible.

There are a few options as to how to control the SDRAM clocks:

- Disable the clock driver chip(s) during testing by negating the `ENABLE` input of the chip.
- Drive the clock by an external signal.
- Drive the clock from an unused FPGA or CPLD pin. The spare pin is tri-stated during normal operational mode and in testing mode is driven by boundary-scan.
- Provide on board jumpers so that the clock to each SDRAM device can be connected to another boundary-scannable pin during testing (and to the clock driver chip during normal operational mode).

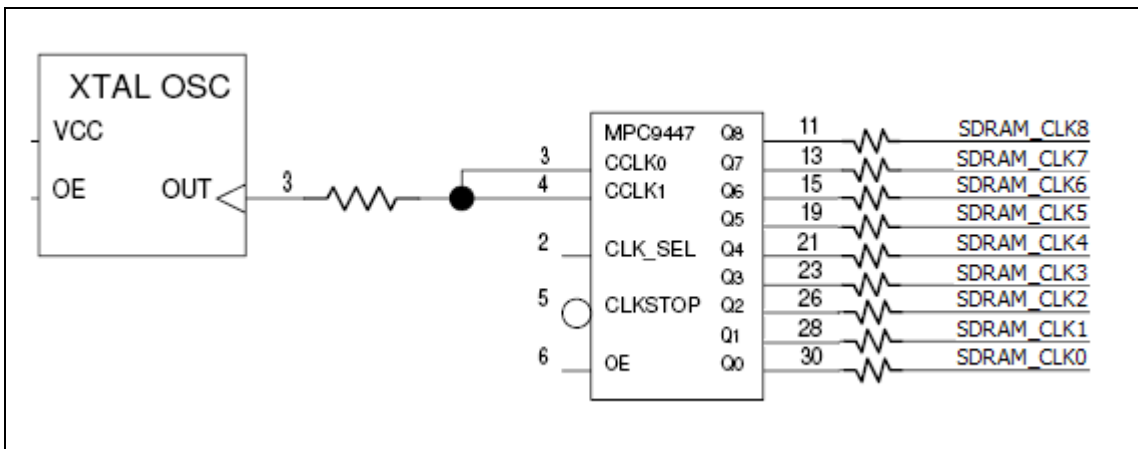


Figure 15: SDRAM clock driver is 'free running'

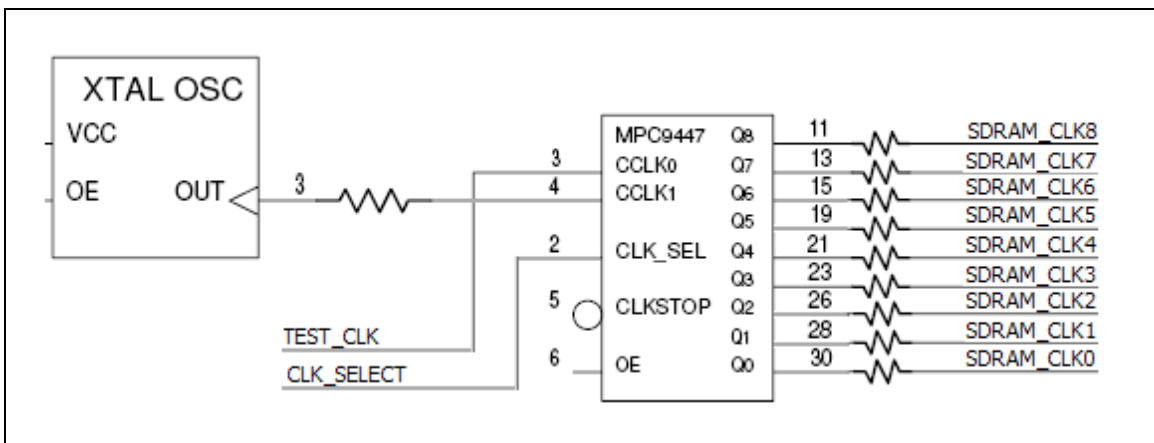


Figure 16: SDRAM clock controlled by 2 boundary-scan signals

If none of the above is implemented in your current design, the memory device cannot be tested using traditional JTAG test. If your design includes a processor, consider the Corelis JTAG Emulation Test (JET) technology as an alternative. ScanExpress JET™ allows you to test some memories that are not testable by boundary-scan methods, and it also supports complete functional at-speed testing of memory and flash devices whether or not they are boundary-scan testable.

Flash Programming

In-system programming (ISP) of flash devices through JTAG is done by emulating read/write cycles to the flash device using the surrounding boundary-scan devices. Therefore, when programming flash devices that do not include boundary-scan capability, it is critical to have the ability to control the pins on the flash. This includes, but is not limited to, address pins, data pins, chip select/enable, output enable, clock (synchronous devices), and write enable. Note that in addition to this, the boundary-scan device(s) that are used to apply patterns to the flash device in-system must have the capability of driving the address and control pins on the memory device, and independently have the ability to drive or sample from the data pins on the flash device. On some boundary-scan devices, multiple outputs on the device are assigned to the same control cell in the boundary-scan register, limiting the way the outputs are configured/controlled. Data pins must never be assigned to the same control cell as other pins that are mandatory for programming the flash device (address, chip select, write enable, etc.).

If there are non-boundary-scan components such as directional buffers between a boundary-scan component and the flash device, the Corelis tool is able to generate patterns to control this buffer to have data flow in the direction as required, as long as the direction signal is controlled by a boundary-scan device.

Design For Test (DFT) Guidelines for Boundary-Scan Testing

Flash Timing Consideration

The following equation can be used for calculating the absolute minimum flash programming time through a JTAG interface (that is, the best theoretical programming time that can be achieved):

$$\frac{(\#bits\ in\ chain) * (\#scans/write) * (\#writes/location) * (\#locations)}{TCK\ frequency}$$

Where:

#bits in chain	Effective length of the boundary-scan chain (assuming unused components placed in BYPASS)
#scans/write	Number of DR scans required to write a data value to the flash
#writes/location	Number of data values that must be written to program each location
#locations	Number of data locations to be programmed
TCK frequency	Frequency of the JTAG TCK signal

Below is a sample calculation, assuming the following conditions:

Device	28F016SV (size is 2Mbyte = 2097152 bytes or 1048576 words)
Data Access Type	Word (16-bits)
TCK frequency	10 MHz
#scans/write	1 (if direct write provided to flash) 2 (otherwise)
#writes/location	2 (command value and then data value)
#bits in chain	701 (2 devices in the scan chain, one BSD with 202 cells that is connected to the flash and another BSD with 499 cells)

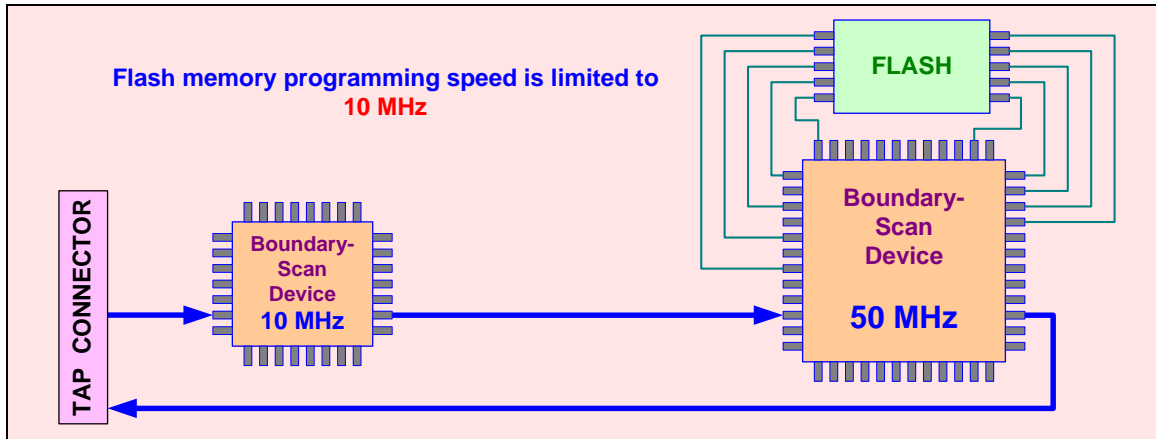


Figure 17: Flash Programming Speed is Limited by the Slowest Device in the Chain

If there are additional devices in the chain and they cannot be placed in BYPASS mode (BYPASS adds 1 bit to the equation for each device), then all the cells are added up to get the total number of cells (e.g., if there are 2 devices, the original device with 202 cells and one with 499 cells). Using these values, the theoretical programming time for programming the entire flash device is:

$$((701 * 2 * 2 * 1048576) / 10000000) = 294 \text{ seconds}$$

The programming time can be reduced by 50% by utilizing the 'direct write' method (connecting the WE-pin of the flash to the TAP connector). In this case, the total programming time would be 147 seconds:

$$((701 * 1 * 2 * 1048576) / 10000000) = 147 \text{ seconds}$$

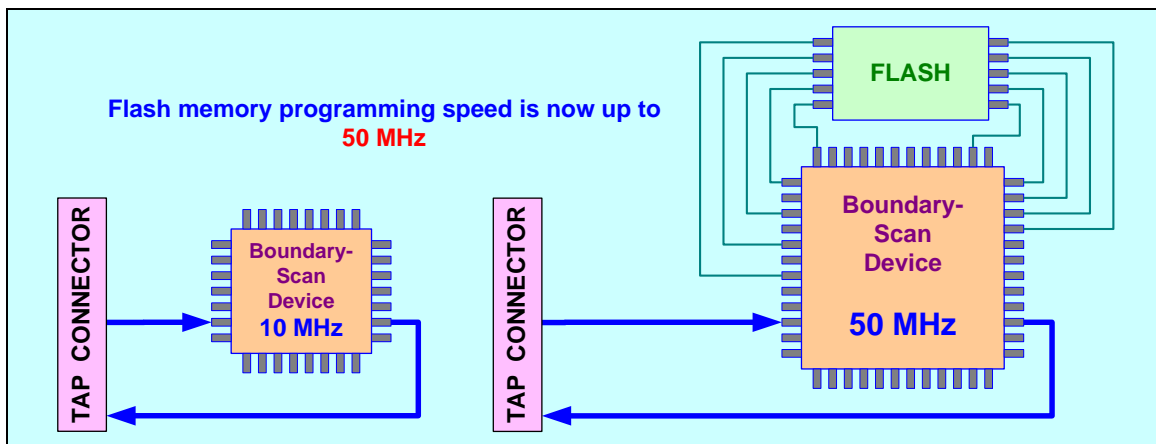


Figure 18: Separate the Scan Chains to Increase Flash Programming Speed

Design For Test (DFT) Guidelines for Boundary-Scan Testing

Large flash devices on the board may take a very long to program. The programming time can be reduced by decreasing the length of the scan chain and increasing the maximum TCK rate. In applications where flash programming speed is important it is beneficial to put the boundary-scan device that connects to the flash on its own separate scan chain. This reduces the effective scan chain length and removes any maximum TCK speed limitations that would be imposed by other boundary-scan devices in the chain. For example, the programming time is significantly improved when the effective scan chain length is reduced to 202 bits and the TCK frequency increased to 50 MHz. Using the same equation, the new time becomes:

$$((202 * 2 * 2 * 1048576) / 50000000) = 17 \text{ seconds}$$

The time calculated is theoretically the lowest possible programming time that can be achieved with boundary-scan. In practice (depending on the speed of the computer used, length of the scan chain, etc.), the actual “real world” programming time can typically vary from anywhere from 1.4 to 3.3 times this value. However, it cannot be lower than the minimum programming time specified by the flash’s manufacturer.

In case of an Intel 28F640J3 using the 32-Byte Write Buffer the Effective Programming Time per Byte is 6.8µs, which translates into 14.3s for a 2MB file (word mode, Intel data sheet # 290667-015, January 2004).

If the target board also includes a microprocessor and it is connected to the flash device, Corelis offers Target-Assisted Flash Programming (TAFP), which uploads a small program to the CPU along with the data and instructs the CPU to do the flash programming. This approach allows near theoretical programming speed. Contact Corelis Support at support@corelis.com for a list of supported CPUs.

Connectors and Other Interfaces

The Corelis Boundary-Scan I/O hardware provides boundary-scan controlled digital I/O that can be used to support testing of board edge connectors, PCI connector, Memory sockets and any other connectors which are accessible, as well as internal board nodes that are otherwise inaccessible through boundary-scan.

To use a Scan I/O module, connect the digital I/O on the Scan I/O module to the nodes on-board that you want to access with boundary-scan and place the boundary-scan devices of the Scan I/O in series with the on-board boundary-scan devices. The Scan I/O boundary-scan devices are added to the board topology, and then Corelis ScanExpress Merge automatically combines the circuit board and the Scan I/O netlists.

Corelis offers special Scan I/O modules:

Model	Description
ScanIO-300LV	Digital I/O controller with 300 Pins
ScanPCI	PCI/PXI/cPCI Boundary-Scan tester
ScanDIMMs	Interconnect tester of DIMM sockets
ScanPlus Chip Tester	Boundary-Scan device tester

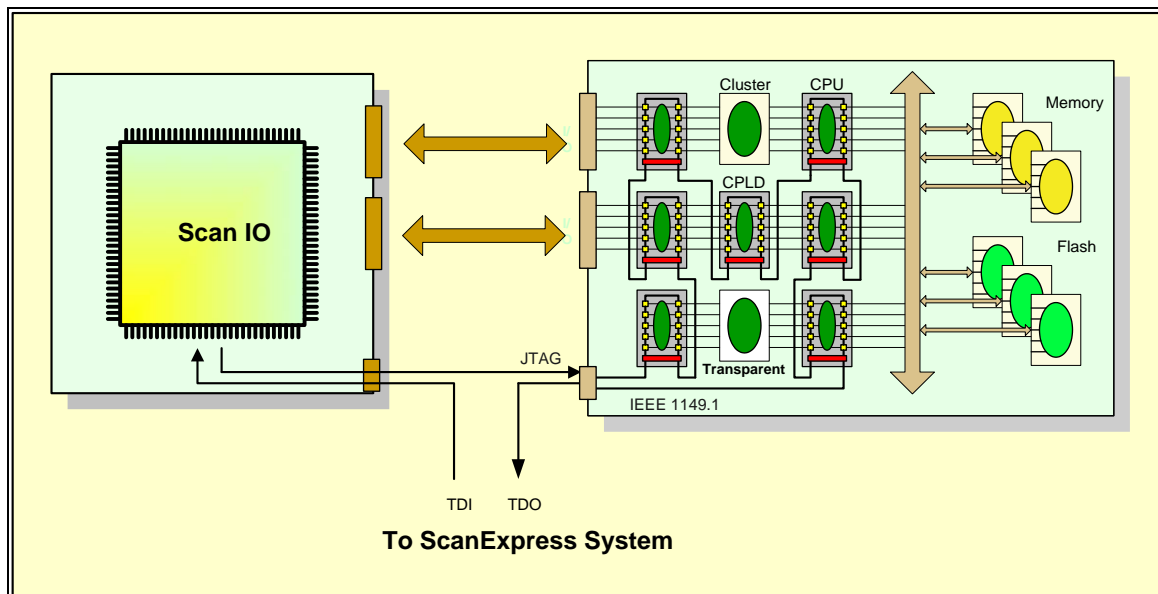


Figure 19: Implementation of a Scan I/O Module

Device-Specific Considerations

Individual devices may be boundary-scan compliant only under certain conditions. In addition, sometimes these conditions are not documented by the device manufacturer. See **Appendix B: Device-Specific Requirements** for a partial list of such devices and information about using them for boundary-scan applications.

Test Expectations

What can be expected from boundary-scan testing? The more boundary-scan devices you have on your board, the higher the chance of better coverage. If there is only one BSD, it cannot check the interconnectivity with other BSDs and therefore the test is limited to checking for shorts most of the time. Only if there is ‘feedback’ from either another BSD or from memory devices does the coverage also include the test for open pins. Feedback can also be accomplished by using ‘loopback’ connections on connectors if the pins of the connector are connected to the BSD. Corelis separates the connectivity tests into the ‘Interconnect’ and the ‘Buswire’ tests. If there is no feedback available, then there is also no Buswire test generated. The Test Coverage Report generated includes the break-down of all NETS as well as all PINs giving the user the most information and details down to the pin level. Each net’s or pin’s testability is categorized as ‘none’, ‘partial’ or ‘full’ coverage. As an example, the Corelis ScanPlus Demo Board has been designed with very high testability and has a total net coverage of 94%, with 79% fully and 15% partially tested. Similar percentages are achieved for the pin-level coverage.

Boundary-scan test is a very basic test for shorts and open pins by toggling the scannable pins and reading their state. If the state is wrong or different than what is expected, the test fails. In the same way – by toggling the pins in a certain sequence – enhancements are implemented to test and program memory devices, including many types of RAM (SDRAM, DDR2, etc.) as well as a variety of programmable parts, including flash, EEPROM, FPGA, and configuration PROM.

All these tests are generated automatically by selecting the right group or individual device within an automated test pattern generator (ATPG). Corelis ScanExpressTPG also allows the user to custom ‘drive’ any scannable pin by introducing a C-like scripting language with a large list of built-in functions. The purpose is not only to control the boundary-scan cells, but also for program control (e.g., loops), decision-making functions, reading/writing to external files, text operations, binary operations and much more. This gives the user the tools to work out any custom solution, if there is a need for it.

At-Speed Functional Testing

As powerful as boundary-scan tests are and as good as the accessibility may be on the particular UUT, it cannot find faults that only show up if the board is running ‘at-speed’. Even though some boundary-scan controllers are capable of driving a test clock of 100MHz (BSDs typically specify TCK at 5-25MHz), the resulting speed of any signal change on a scannable pin would still be in the kHz range. Therefore, no cross-talk or noise related problem would show up at this type of speed. To complement boundary-scan testing with real-time testing, Corelis offers JTAG Emulation Test (JET) technology.

Corelis ScanExpress JET is JTAG Emulation Test technology, which complements traditional JTAG boundary-scan tests with the power of the CPU in real time run mode. Any target, which uses a CPU with a JTAG emulation port, allows the user to utilize this CPU for testing purposes.

ScanExpress JET extends the test coverage beyond the shorts, open pins, and basic function tests of traditional boundary-scan test steps by executing all instructions at full CPU clock, which means that faults based on noise or cross-talk can be detected. Those types of faults would otherwise go undetected with regular boundary-scan technology.

In some instances, ScanExpress JET is the only feasible alternative for test. For example, the clock signals to SDRAM memory are not controllable by boundary-scan cells so there no boundary-scan memory test is possible. But with JET technology, not only can the memory be tested, but since the test is executed at CPU speed, the entire memory address space can be tested. Testing the entire memory space with traditional boundary-scan tools would take hundreds of times longer than the same test executed with ScanExpress JET.

Test steps generated with Corelis ScanExpress JET are fully compatible with the Corelis test executive, ScanExpress Runner, with applications in both the prototype and production environment.

Visit the Corelis website at <http://www.corelis.com/products/ScanExpressJET.htm> for more information about ScanExpress JET.

Additional Testing

Many designs have mixed analog and digital signals or have areas that cannot be tested with boundary-scan tests. Some exceptions are digital-to-analog converters (DACs) and analog-to-digital converters (ADCs) when both can be accessed by BSDs using the Corelis scripting language to configure a setup, analyze the result and compare it with valid values. Based on that comparison (all done within the script), a PASS or FAIL of this test step can be determined. In many cases, when analog or discrete components are involved, it is unavoidable to use traditional resources in order to find defects as early as possible in the process. Nets which are not covered by the Corelis boundary-scan tools are reported separately. Using these special output reports allows the user to manufacture a fixture with a smaller set of nails reducing the cost for the fixture manufacturing and fixture maintenance.

Corelis was successfully implemented in a variety of BON (Bed Of Nails) test equipment and Flying Prober. Using the ScanExpress application programming interface (API), calls to boundary-scan tests can be executed from any third party executables, making the boundary-scan test steps part of the In-Circuit Tester (ICT) execution. By eliminating the need for another test station and additional board-handling, customers are able to speed up their production process and gain very high test coverage. All of this can be accomplished by incorporating Corelis boundary-scan tools into the overall test strategy.



Figure 20: Flying Prober and BON testers

Finding Faults: Now What?

Having high test coverage also means that during testing you will find many faults which happen during the manufacturing process. While of course this is a good thing on one side, analyzing the results and locating the faults on your board is yet another issue. This part has nothing to do with a good DFT practice, but since Corelis helped during the design and the test phase of the product, we also have tools for the repair. ScanExpress Advanced Diagnostic Option (ADO) analyzes the results from ScanExpress Runner (such as a short between two nets) and gives you likely locations by means of listing pins which are on adjacent components. Making use of ScanExpress Viewer also allows you to visually see the location of faults on a screen, showing the CAD locations of component pins overlaid by a picture of the board for both the top and the bottom side, as in Figure 21.

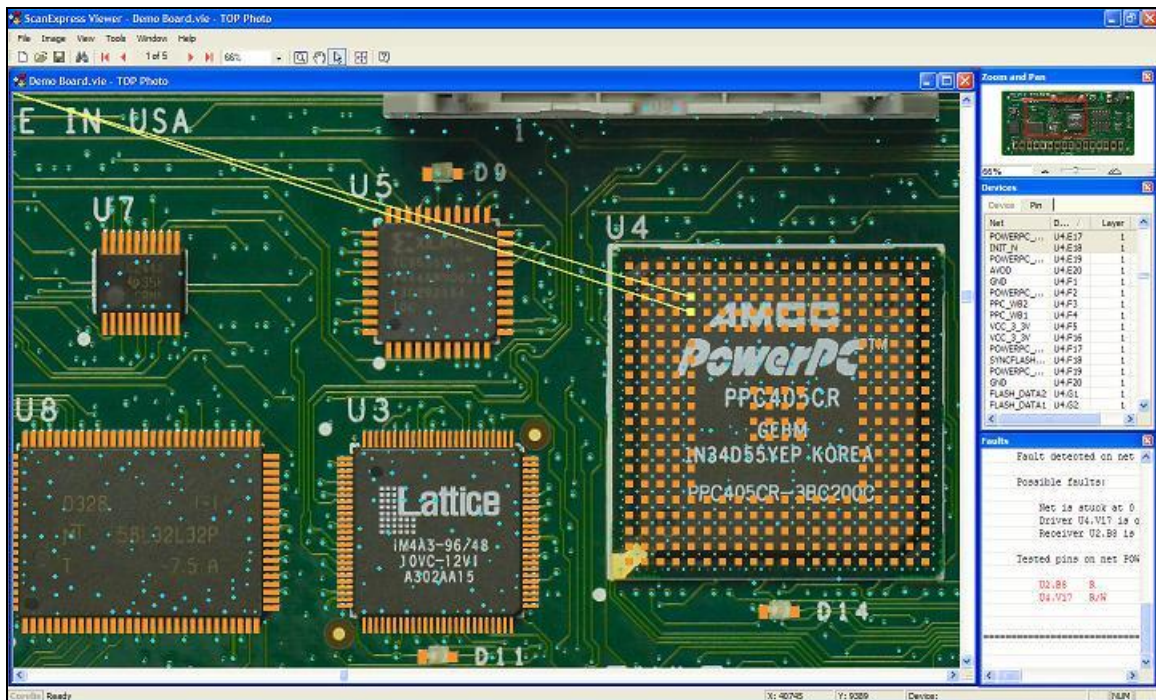


Figure 21: Corelis ScanExpress Viewer showing isolated faults

Appendix A: Glossary of Abbreviations

ADC	Analog-to-Digital Converter
AOI	Automated Optical Inspection
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
ATPG	Automated Test Pattern Generator
AW	AutoWrite
BDM	Background Debug Mode
BGA	Ball-Grid-Array
BSD	Boundary-Scan Device
BSDL	Boundary-Scan Description Language
BSR	Boundary-Scan Register
CF	Compact Flash
CPLD	Complex Programmable Logic Device
CS	Chip Select
DAC	Digital-to-Analog Converter
DDR	Double Data Rate (as in DDR SDRAM, DDR2, DDR3 are newer, larger and faster)
DFT	Design for Test (or Testability)
DIOS	Digital Input Output Scan
DNI	Do Not Install
DNP	Do Not Populate (or Place)
DSP	Digital Signal Processor
ECL	Emitter Coupled Logic
EEPROM	Electrical Erasable Programmable Read Only Memory
FAT	First Article Test
FPGA	Field Programmable Gate Array
Hi-Z	High Impedance
I/O	Input – Output
IC	Integrated Circuit
ICT	In-Circuit Test (or Testing)
IEEE	Institute of Electrical and Electronic Engineers
IP	Intellectual Property
ISP	In-System Programming
JEDEC	Joint Electron Device Engineering Council
JTAG	Joint Test Action Group
LSB	Least Significant Bit
LSP	Local Scan Port
LVTTL	Low Voltage TTL (Transistor-Transistor-Logic)
MCGR	Multi-Cast Group Register
MCM	Multi-Chip Module
MDA	Manufacturing Defect Analyzer
MSB	Most Significant Bit
OE	Output Enable
PCB	Printed Circuit Board (also PWB)
PCBA	Printed Circuit Board Assembly (PCB with components installed)
PLL	Phase Lock Loop
PROM	Programmable Read Only Memory
PWB	Printed Wire Board (also PCB)
RAM	Random Access Memory
ROM	Read Only Memory
SDRAM	Synchronous Dynamic Random Access Memory
SI	Scan Input
SO	Scan Output

Design For Test (DFT) Guidelines for Boundary-Scan Testing

SRAM	Synchronous Random Access Memory
SSRAM	Synchronous Static Random Access Memory
TAP	Test Access Port
TCK	Test Clock
TDI	Test Data Input
TDO	Test Data Output
TMS	Test Mode Select
TPG	Test Pattern Generator
TRST	Test Reset (TRST* active low)
TTL	Transistor-Transistor-Logic
USB	Universal Serial Bus
UUT	Unit Under Test
VLSI	Very large Scale Integration
WE	Write Enable

Appendix B: Device-Specific Requirements

The IEEE-1149.1 standard describes in detail the operation of components that comply with the standard. There are some cases where components comply with the standard only under specified conditions or are not compliant at all.

The following is a compilation of devices, listed alphabetically by manufacturer, we found that need special attention. However, as lengthy this list may be, there is no claim that it is complete or contains the most up-to-date information. For additional information, contact Corelis Support at support@corelis.com. In addition, if you believe that you are aware of corrections or updates to this information, please contact Corelis Support at support@corelis.com.

Actel

Actel Boundary-Scan Testing and Programming Info

- Some devices require elevated voltages to program. Note that this is sometimes difficult to find. Check:
 - The data sheet text.
 - The device parameters section, possibly the "Recommended Maximum Operating Conditions Programming and PLL Supplies" Table.
 - "Vpump", "Vpn" and "Vpp".
- We have heard of parts that require elevated programming voltages failing infrastructure after attempted programming. This appears to be due to device current requirement spikes, causing voltage drops and possibly voltage ringing seen at the device. Recommend providing adequate bypass capacitors (this is recommended in the Actel data sheet) and low impedance wiring.
- A boundary-scan controller with a constant clock is required. A PCI-Turbo with ScanTAP-4 and NetUSB-1149.1/E meets this requirement.
- Corelis tools directly accept the Actel STAPL (.STP) file format for programming.
- Some Actel FPGAs do not support boundary-scan testing until programmed. Actel offered pre-programmed devices from the factory at one time. So, for a short period of time, the initial programming by the user was not necessary. This has been discontinued.
- We have experienced that for some devices, the Actel provided programming and BSDL file "set" do not work.
- In some cases, modifications to the STAPL file were required.

Design For Test (DFT) Guidelines for Boundary-Scan Testing

- The programming power pin may be connected only to the programming I/O connector that the Actel programming controller plugs into (and maybe some capacitors). Therefore, even if the programming voltage is not an elevated voltage, there may still not be any power provided to the programming power pin.

Actel 54SX Family FPGAs

(Applies to A54SX08, A54SX16, A54SX16P and A54SX32).

From Actel data sheet v3.0.1 May 2000:

“In the dedicated test mode, TCK, TDI and TDO are dedicated pins and cannot be used as regular I/Os. In flexible mode, TMS should be set HIGH through a pull-up resistor of 10k. TMS can be pulled LOW to initiate the test sequence. The program fuse determines whether the device is in dedicated or flexible mode. The default (fuse not blown) is flexible mode.”

This is not listed as a compliance enable attribute in the BSDL file. This should not normally be a problem, especially if the Corelis TAP recommendations are followed. A scenario is if TMS must be high from power up (the data sheet was not specific on this), the device is in flexible mode, there is no pull-up on TMS in the design, and the board is powered up without the TAP cable connected. The device may not go into boundary-scan mode even after the TAP cable is connected.

Altera

Altera EPM7128ATC and Similar Parts

While in boundary-scan mode, the device pins are supposed to be fully controlled by the boundary-scan logic. This is not the case with some devices from Altera. AT device (specifically the EPM7128ATC) pins will take on the programmed type during JTAG operations. What this means is that if a pin is programmed as open-drain, the pin will remain open-drain even during JTAG operations. Furthermore, Altera pointed out that there is an option to automatically convert (optimize) pins to open-drain if they do not drive.

In this case the programmed part does not fully comply with the (generic) BSDL file for the device. The simplest solution to the problem is to erase the device prior to executing interconnect or cluster tests.

Altera FLEX 6000 Parts

The Altera FLEX 6000 family is fully boundary-scan compatible in the pre-configuration mode. During configuration, the devices are NOT boundary-scan compatible. After configuration, the devices may or may not be boundary-scan compatible, depending on the configuration data option that was selected by the user (when preparing the configuration data using the Altera development tools).

In order to perform boundary-scan testing on FLEX 6000 devices, the `nCONFIG` pin of each device must be held LOW (“0”). The `nCONFIG` pin is a boundary-scan compliance enable pin that must be held at “0” to enable boundary-scan testing and kept “0” during the entire boundary-scan test.

Freescal e / Motorola

Freescal e/ Motorola MC68306

BSDL file error: Pins using output control cell 67 have “0” in the safe-disable field. This is incorrect. It should be “1”.

Example:

Incorrect:

```
"89 (BC_2, OEB, output3, X, 67, 0, Z)," &
```

Corrected:

```
"89 (BC_2, OEB, output3, X, 67, 1, Z)," &
```

This should be changed for the many pins that reference output control cell 67.

Observation: The safe disable field is not the same value for all pins. Some pins use “1”; some pins use “0”. This is certainly within the IEEE-1149.1 specification, but this is unusual and makes it more difficult to ascertain whether this is the only output control cell this error applies to.

This has not been corrected in the Freescal e/Motorola distribution BSDLs as of July 2007. This is the only such error found in the board we worked with. It is unknown if this applies to any other members of this family.

Freescal e/ Motorola MCF527x ColdFire (and possibly others in this family)

Note:

This does not apply to MCF5272. We only found the two signals that are already compliance enables in the BSDL file, `TEST` and `JTAGEN`.

Compliance enable `HIZ` pin must be high.

TAP pin purpose selection pin: `MTMOD`, low for boundary-scan, high for emulation. From MCF5272 ColdFire Integrated Microprocessor User’s Manual, MCF5272UM/D, 3/2002, REV 2:

19.17.6 Motorola Test Mode Select (MTMOD)

`MTMOD`: When the `MTMOD` input is low, `JTAG` mode is enabled. When it is high, `BDM` mode is enabled.

Freescall/Motorola MCF547x ColdFire

There are no compliance enables listed in the "MCF5475BSDL.bsd" BSDL file.

From "MCF5475RM.pdf", MCF547x Reference Manual. Rev. 2.1. 10/2004
22.2.1.1 Test Mode 0 (MTMOD0):

The MTMOD0 pin selects between Debug module and JTAG. If MTMOD0 is low, the Debug module is selected; if it is high, the JTAG is selected. Table 22-2 summarizes the pin function selected depending upon MTMOD0 logic state.

Freescall/Motorola MPC8540 PowerQUICC III

Summary:

- Compliance enables: Pins AG19 and AH20 must be high.
- TRST* and HRESET* must be driven low during power up at the same time.
- Based on past experience with other Freescall/Motorola processors in this family, this device probably requires HRESET* to be high to enable boundary-scan mode.
- Assert TAP_EN. (External pin not found. Must be an internal pin on the Core.)

From BSDL file "MPC8540.bsd":

```
attribute COMPLIANCE_PATTERNS of MPC8540: entity is  
"(LSSD_MODE_L, TEST_SEL) (10)";
```

This means that pins AG19 and AH20 must be high.

Freescall/Motorola MPC8545E, MPC8548E PowerQUICC III

MPC8548E and MPC8545E have a problem which requires an unusual workaround. This may apply to other PowerQUICC devices.

From Device Errata for the MPC8548E PowerQUICC III Processor (MPC8548ECE Rev. N) PDF, page 227:

If the SAMPLE/PRELOAD or EXTEST instruction is the current instruction in the JTAG TAP, and the JTAG state machine is moving into the UPDATE-DR state, TMS must be held pass the falling edge of TCK.

Recommended workaround:

Hold the TMS signal 2ns past the falling edge of TCK by adding cable length ONLY to the TMS signal. It is recommended to also extend a close-by ground wire. Successful cable lengths vary from 2 ft. to 8 ft.

Motorola MPC755

From MPC750UM/D, 12/2001, Rev. 1:

Test reset (TRST) is a JTAG optional signal which is used to reset the TAP controller asynchronously. The TRST signal assures that the JTAG logic does not interfere with the normal operation of the chip, and must be asserted and de-asserted coincident with the assertion of the HRESET signal.

Motorola MPC74xx

From: MPC7450UM/D, 2/2003, Rev. 3, “MPC7450 RISC Microprocessor Family User’s Manual”.

TRST: Asserting this input causes asynchronous initialization of the internal JTAG test access port controller. Note that the signal must be asserted during the assertion of HRESET in order to properly initialize the JTAG test access port. The TRST signal must be asserted to properly initialize the boundary-scan chain. This may be accomplished by connecting it to HRESET, using logic to OR any external JTAG TRST drivers.

Note that this input contains an internal pull-up resistor to ensure that an un-terminated input appears as a high signal level (negated) to the test logic.

Motorola MPC8245

From MPC8245CE, Rev. 9, 1/2004, MPC8245/MPC8241 Integrated Processor Chip Errata:

The MPC8245/MPC8241 may experience problems on the TBEN, SRESET, TRIG_IN, and CHKSTOP_IN signals with the HIGHZ, EXTEST, and CLAMP JTAG operations and with the LSRL scan chain and HIZ COP instruction.

For Rev. A devices, the recommended workaround is to modify the board level boundary-scan test program. (This solution may or may not be workable, depending on the testing system). Otherwise, this situation has no work around.

This problem is reported to be fixed in Rev. 1.2 = Rev. B devices.

From MPC8245R2UMAD, Rev. 2.2, 2/2004, MPC8245 Integrated Processor User’s Manual Rev. 2 Errata

Latches that are dedicated to JTAG functions are not initialized during system reset. The IEEE 1149.1 standard prohibits the device reset from resetting the JTAG logic. The JTAG reset (TRST) signal is required to reset the dedicated JTAG logic during power-on.

Motorola MPC555

From the User’s Manual (c22jtag.pdf, Rev. 15 October 2000):

22.2 JTAG Signal Descriptions

To enable JTAG on reset for board test, bit 11 (DGPC select JTAG pins) and bit 16 (PRPM peripheral mode enable) of the reset configuration word should be held high during the rising edge of reset (see 7.5.2 Hard Reset Configuration Word). These need to be configurable on the user board to allow JTAG test of a board. To allow normal operation of the board, these bits need to be low in the reset configuration word.

22.3 Operating Frequency

The TCK frequency must be between 5 MHz and 10 MHz. This pin is internally driven to a low value when disconnected.”

Motorola MPC823

From "MPC823 Design Checklist (MPC823SI1).pdf"

- Connect TRST~ to PORESET~ through a diode, pointing towards PORESET~.
- Driving the data bus with an active device instead of pull-ups during reset configuration ensures that the appropriate value is written into the configuration register. If you do or do not use pull-ups on the data bus and default hard reset configuration word is acceptable, tie the RSTCONF* pin high. If you want to change the hard reset configuration word to something other than the default, drive the appropriate data bus lines low with an active device and connect the RSTCONF* pin to ground. If the default hard reset configuration word is not acceptable and power consumption is a concern, drive the appropriate data lines high with an active device and connect the RSTCONF* pin to ground.

From "MPC860 Board Design Checklist (AN2100).pdf"

Hard Reset Configuration Word Pins

- The data bus is sampled at reset to determine the initial configuration of the MPC8xx if the RSTCONF* pin is tied low. Ensure that the correct values are driven to the bus during the PORESET and HRESET periods. For any data pins that the user wishes to configure as 1's (i.e. where the default configuration is not appropriate), do not expect pull-ups to be sufficient. Use active drivers to drive 1's and 0's, if pull-ups are already being used on the data bus. The drivers should then be disabled following reset.
- Keep HRESET* inactive high after power up.

From "PowerPC MPC823 Reference Manual (MPC823UM).pdf"

4.3.1.1 HARD RESET CONFIGURATION WORD.

Data Bus Bits 11, 12 (counting from zero) DBPC

Debug Port Pins Configuration

11 = ALE_B/DSCK/AT1 functions as DSCK.

IP_B6/DSDI/AT0 functions as DSDI.

OP3/MODCK2/DSDO functions as DSDO.

IP_B7/PTR/AT3 functions as PTR.

TCK/DSCK functions as TCK.

TDI/DSDI functions as TDI.

TDO/DSDO functions as TDO.

- Watchdog timer
 - The Watchdog timer is not disabled in boundary-scan mode. It is disabled in emulation mode. This will cause periodic resets.

Design For Test (DFT) Guidelines for Boundary-Scan Testing

- This alone may interfere with boundary-scan testing.
- It will also re-load the HRCW. If D11, D12 are not reloaded correctly, the device may exit boundary-scan mode. Interference on D11, D12 could come from other devices, boundary-scan and non-boundary-scan, driving the D11, D12 nets.

Motorola MPC860

The Motorola MPC860 supports on-chip debugging through a BDM port and also boundary-scan testing through a JTAG TAP. The MPC860 is designed such that the BDM and JTAG functions are shared on the same pins, the function of which is determined at power up. When using the MPC860 for boundary-scan testing, you need to ensure that the chip is configured such that the shared pins are functioning as JTAG TAP and not BDM. Refer to the MPC860 documentation for additional details.

Next, the pin functions can be switched (overridden) by writing to an internal MPC860 register (DBPC field in the SIUMCR register). Therefore, it is possible that software executing on the MPC860 may change the function of the pins, even though they are JTAG at power up.

The boundary-scan TAP is set to Boundary-Scan during reset. There are three requirements:

1. Set the DBPC bits [12,11] to [1,1] in the Hard Reset Configuration Word (HRCW). This can be done by setting the MPC860 data bus bits 12 and 11 high during a chip reset. One easy way to implement it is to provide pull-up resistors on the D11 and D12 signals. This way every time that the HRESET# is active (low), the data bus is tri-stated and the signal level on these lines will be "1" because of the pull-up resistors.
2. In addition, the RSTCONF# signal must be active (low) during chip reset to enable the automatic writing of the data bus content into the Hard Reset Configuration Word whenever the chip is being reset (HRESET# asserted, then de-asserted).
3. HRESET# must go inactive with the proper timing relative to PORESET# signal.

For more details, please refer to the MPC860 PowerQUICC User's Manual, MPC860UM/AD. In the 07/98, REV. 1 version, see Section 12.3 ("MPC860 Reset Configuration"), Table 3-1 ("Signal Descriptions" for a description of the RSTCONF# AND HRESET# pins), Section 13.3 ("Internal Pull-Up and Pull-Down Resistors, Section 13.4.1 ("Reset Configuration"), and Table 12-3 ("Hard Reset Configuration Word Field Descriptions" for a description of the DBPC bits).

With the HCRW set for Boundary-Scan mode, the following multi-purpose pins will be set as noted:

- TCK/DSCK functions as TCK
- TDI/DSDI functions as TDI
- TDO/DSDO functions as TDO

Note that the setting of the two data bus bits is different for BDM mode and JTAG mode. If both are to be used, the board designer will need to accommodate both situations. Motorola has recommendations on how to implement that.

Remember that if the data bus is not being used explicitly to define the Hard Reset Configuration Word, the default for this word is “0”, which means that DBPC bits are not configured for JTAG operation.

Motorola MPC8260, MPC8270 PowerQUICC 2 Family

The Motorola MPC8260 supports a compliance enable pin that is not properly mentioned in the BSDL file. The `PORESET*` pin of the Motorola MPC8260 device must be high in order for boundary-scan to work properly. This pin must be held high BEFORE and DURING boundary-scan testing by the circuit of the board under test or by forcing the pin to the correct state with external means such as jumpers, probing, etc.

Note that a constraint statement in a Constraints file cannot be used to set a compliance enable pin to the correct value BEFORE the test. It can only be used to maintain the signal at the correct state DURING the test. Therefore, even when the `PORESET*` pin is connected to a boundary-scannable output pin and this output pin is specified as a `FIXED_HIGH` constraint in the `ScanExpressTPG` Constraints file, it is not enough because the `PORESET*` pin MUST be set to the HIGH state even BEFORE the boundary-scan test commences.

Motorola 56303

Earlier versions of the Motorola 56303 DSP do not support the `SAMPLE/PRELOAD` instruction. The following DSP56303 silicon masks contain bugs that make them non-compliant with Boundary-Scan IEEE Std 1149.1:

- OF88S
- OF94R
- 1F94R
- OJ22A
- 2J22A
- 3J22A
- 4J22A

The following DSP56303 does not list boundary-scan non-compliance in its errata. However, you may want to verify this with Motorola before purchasing:

- OH82G

IDT

IDT 64474/64475/64574/64575 Microprocessors

In order for the IDT 64474/64475/64574/64575 processor chips to be fully IEEE-1149.1 compliant and operate according to the BSDL description provided by the manufacturer, the following inputs must be setup as described below:

- JTAG32 signal must be connected to GND.
- MASTERCLOCK needs to be running during JTAG operation.
- VCCOK* has to be asserted (LOW) during JTAG operation.

Intel

Intel SA-110

Per the BSDL file:

Ensure to drive PWRSLP to logic 1, or else the chip will sleep!

This is in the design warning section of the BSDL file. It is not in the compliance pattern section. We recommend that this pin be low from power up and all through boundary-scan testing.

Intel SA-1110

Per the BSDL file:

Ensure to drive BATTF and VDDF to logic level 0 or else the chip will sleep!

This is in the design warning section of the BSDL file. It is not in the compliance pattern section. We recommend that these two pins be low from power up and all through boundary-scan testing.

Intel 82543

The compliance enable for I_TRI_OUTPUTS_N may be commented in the BSDL file. It was found to be necessary in one of our test procedures. Keep this pin low to allow the PCI bus pins to operate correctly in boundary-scan.

Lattice

Lattice Non-Compliant Parts

There is a known chip problem with some Lattice parts (e.g., ispLSIGDX160V BGA272). The value of the IR capture of this chip is “10110”, which is not IEEE-1149.1 compliant. Lattice fixed this problem on production parts and it only exists on engineering samples.

To deal with the non-compliant devices:

1. Use ScanExpressTPG and ScanPlus Runner versions 3.45 or later, or any ScanExpress Runner version.
2. Use the correct BSDL file (10110) when generating the infrastructure test.
3. Use the Modified BSDL file (11001) for generating all other tests.
4. When using Runner, under the PROPERTIES menu, select the option of "Disable Scan Integrity checks".

Lattice's ScanPath Linker Chip, and other ScanPath Linker devices

The Lattice Semiconductor LSC BSCAN-1 Multiple Scan Port Addressable Buffer ("ScanPath Linker") device is a low voltage second source alternative to the TI 8997 and TI 8996 ScanPath Linker device. Customers often use this device to segment large target boards into several local scan chains where each chain can be enabled or disabled by ScanExpressTPG software. This is especially useful for target boards that have multiple configurations with optional parts and optional daughter cards. The user can remove an option without having to worry about "breaking" the overall card scan chain. The following two figures show a typical implementation of the Lattice ScanPath Linker parts in a 12 TAP (chain) configuration:

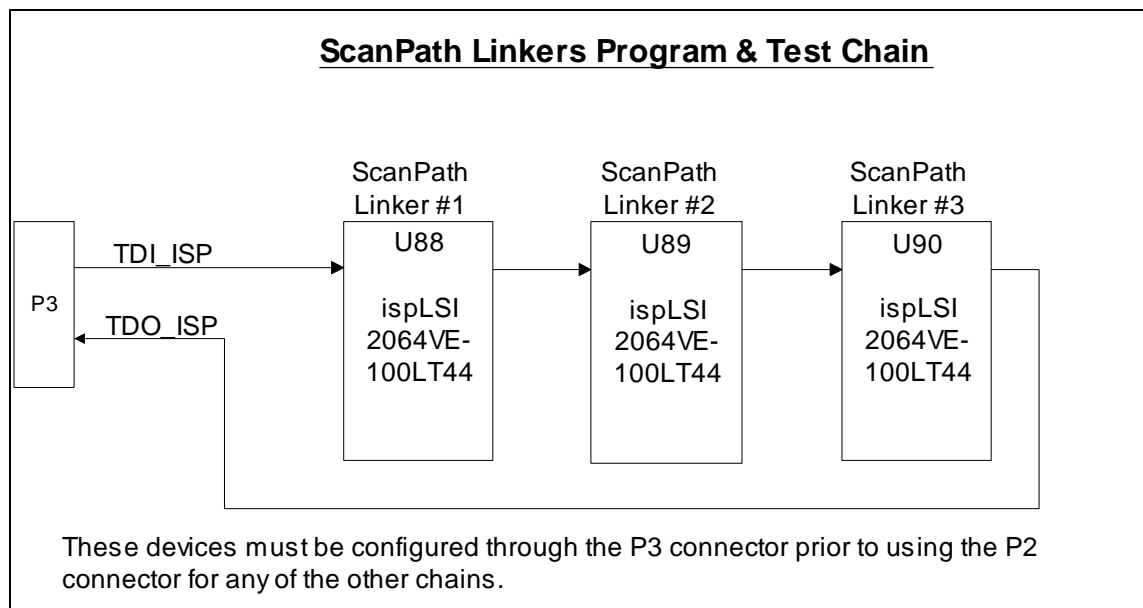


Figure 22: One scan chain using multiple ScanPath Linker

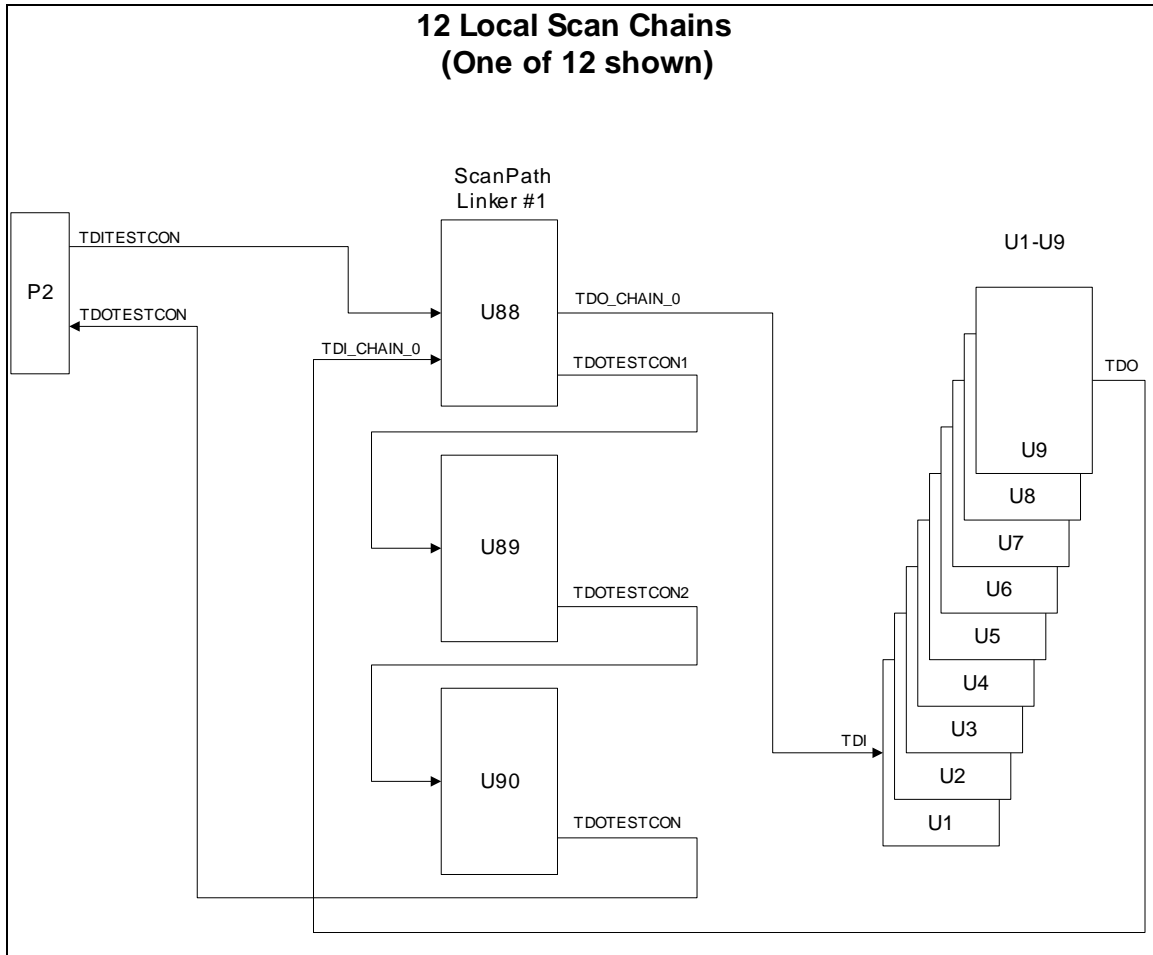


Figure 23: Another Sample of ScanPath Linker

In order to improve the local TAPs signal quality, make sure to program the Lattice CPLD parts with the slew rate limited I/O pins option.

Lucent

Lucent ORCA FPGAs

Lucent ORCA FPGA devices use the PRGM pin as a TRST. However, it is not called, “PRGM/TRST”; it is simply called, “PRGM”. In the description, it mentions that asserting PRGM also resets the TAP controller. Any time PRGM goes low during a test, it resets the ORCA’s JTAG port. Therefore, it is important to be careful and exclude it from testing so it does not accidentally go low.

PLX Technology

PCI9030

There is a known silicon problem with the PCI9030 device. During infrastructure test (IDCODE test) and while being placed in BYPASS mode, this device outputs a data bit

Design For Test (DFT) Guidelines for Boundary-Scan Testing

of “1” instead of a “0”. The IEEE-1149.1 standard specifies that whenever a device is placed in BYPASS mode it must enable its BYPASS register. The BYPASS register must be one bit long and the value of the BYPASS register must be equal to “0”.

While the chip is not boundary-scan compliant, it is otherwise fully functional.

The solution is to place this device in BYPASS mode during infrastructure test only. During all other tests it does not have to be put in bypass and it will work. You put it in BYPASS by specifying YES in the topology file. Note that this topology file fix is only required for the infrastructure test and all other tests do not require this fix.

Texas Instruments

TI 320C6202 DSP

The TI 320C6202 DSP chip has two modes of operation: emulation mode and boundary-scan mode. In order to run boundary-scan interconnect testing, the chip **MUST** be in boundary-scan mode. When EMU0/EMU1 are pulled high, the device is in emulation mode and the boundary-scan Instruction Register (IR) is 8 bits long. If the EMU0/EMU1 pins are pulled low, then the device is in boundary-scan mode and the IR register is 4 bits long (as specified in the BSDL file).

For boundary-scan testing, the user needs to make sure that the /TRST pin is pulsed from high to low while the EMU0/EMU1 emulation pins are pulled low. Also, the user must make sure that at least one clock pulse occurs on the TCK signal while the EMU0/EMU1 signals are pulled LOW in order for the device to enter boundary-scan mode.

TI MSP430 Family of DSPs

The TI MSP430 family of low power DSP chips includes many varieties of parts, including parts that contain internal flash memory. While these parts are **NOT** boundary-scan compatible, these parts incorporate a standard IEEE-1149.1 port that is used for flash memory programming as well as for software debug.

The Corelis ScanExpress flash tools can be used to program MSP430 parts. For example, the MSP430F149 part was programmed on a standalone TI evaluation kit board. In this configuration, the MSP430F149 chip was the only chip in the scan chain.

The following restrictions and suggestions should be noted by the board designer in order to be able to in-system program the flash memory on the MSP430 chips:

- MSP430 series DSP device must be the first chip in the scan chain. Additional boundary-scan compatible devices can only be placed after the MSP430F149 device and not before it. In other words, the TDI signal of the MSP430F149 must be connected to the boundary-scan header connector that goes to the boundary-scan tester.

Design For Test (DFT) Guidelines for Boundary-Scan Testing

- The hardware reset signal must be toggled prior to the start of the Flash programming in order to reset the MSP430 series chip and bring it to a known state. The user can simply connect the reset input of the MSP430 series chip to the ScanExpress Flash ‘External Write’ signal or alternatively manually reset the chip before starting the programming session.
- Connecting the reset signal to the boundary-scan tester is highly recommended. See the Corelis Application Note, ScanPlus Flash TAP, 99-105, for further information on the recommended 16-pin header configuration.

TI TMS320VC5441 DSP Chip

Boundary-scan test requires control of the five test access port signals (TMS, TCK, TDI, TDO and TRST) as described in IEEE standard 1149.1. Two additional signals, EMU0 and EMU1/OFF, are used by TI DSPs to provide emulation debug capability through the JTAG test access port. Also, TI uses these signals for scan-based factory tests.

During boundary-scan tests, EMU0 and EMU1/OFF must be held high while TRST is transitioned from low to high. This operation sets the correct internal test mode for boundary-scan test to be performed. EMU0 and EMU1/OFF should be pulled high through a 4.7k ohm pull-up resistor on each pin. The pull-up resistors are connected to the DVDD power supply for the VC5441.

Boundary-scan ATPG tools should be configured to cycle TRST prior to beginning boundary-scan tests to ensure that the device is in the proper test mode.

Xilinx

Xilinx XC4, XC5, XCS and XCV

There is an IEEE-1149.1 compatibility issue with the Xilinx FPGA chips that requires special considerations. The Xilinx FPGAs (XC4000, XC5000, XCS-Spartan and XCV-Virtex parts) are FULLY boundary-scan compatible only in the following two configuration modes:

1. Before the chip is configured, provided that the INIT* signal was constantly held at “0” thereby inhibiting configuration.
2. After configuration, if boundary-scan is included by the user in the FPGA design (except Virtex parts which are always FULLY boundary-scan compatible after configuration is done).

Vantis

And other devices that do not include a boundary-scan register

Vantis MACH111

Some devices include a JTAG TAP for purposes of in-system programming, but do not include a boundary-scan register for testing. Sometimes you can determine this from the BSDL file for the device or from the datasheet for the device. The Vantis MACH111 falls into this category.

When generating test vectors for your board using ScanExpressTPG, the “compliance problem” field for this device in the board topology file should be set to “YES”, indicating that the device should be placed in BYPASS mode and not involved in boundary-scan interconnect and cluster testing.

Multi-Chip Modules

Boundary-scan compatible multi-chip modules (MCMs) are devices which combine multiple boundary-scan devices onto one module. From the outside it looks like a single device while inside there are multiple chips that are all interconnected on a common substrate and then packaged into a module. No BSDL file can describe such a device as it is a chain of multiple devices, each with its own BSDL file.

The solution is to approach this as if the multi-chip module pins are a connector and the MCM is a little plug-in board. Create a netlist or schematic that connects each boundary-scan device to the appropriate pin on the MCM. Create the topology file for the multi-chip module. Use ScanExpress Merge to combine the multi-chip module with the original design.

An example of such a module is the Dallas DS21FF42, which contains four DS21Q42 boundary-scan devices on one multi-chip module.

Contact information

Corelis, Inc.
13100 Alondra Blvd.
Cerritos, CA 90703
USA
Telephone: +1 562-926-6727
FAX: +1 562-404-6196
Email: info@corelis.com
Web: <http://www.corelis.com>