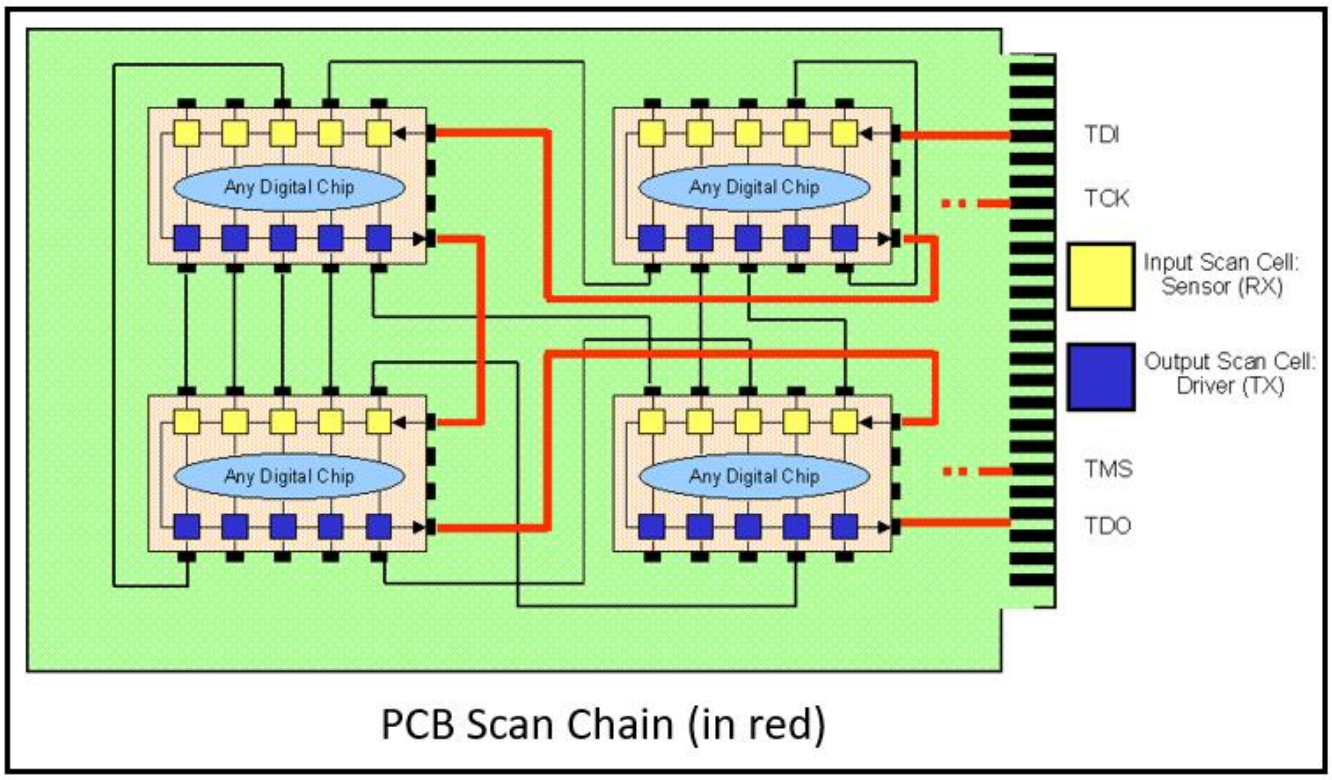
Guidelines for Board Design for Test (DFT) based on Boundary Scan



PCB Scan Chain (in red)

Michael R. Johnson – Product Manager

Michael R. Johnson serves as Product Manager for ScanWorks Boundary Scan Test (BST) for ASSET InterTech, Inc. He also serves as manager of ASSET's Application Engineering organization. As Product Manager, Michael is responsible for providing strategic direction to ASSET's ScanWorks Test Platform by listening to customers and working with cross-functional teams to deliver solutions to meet their needs.

Prior to joining ASSET in 2007 as a Sr. Applications Engineer, Michael's background included roles as a Cellular System Engineer with Nortel Networks and a Hardware Design Engineer with Alcatel USA. While at Alcatel USA, Michael's emphasis was printed circuit board design and layout for high-speed digital cross connect and fiber optic systems.

Michael earned a Bachelor of Science degree in Electrical Engineering from Southern University and A&M College located in Baton Rouge, Louisiana, and a Master of Business Administration with an emphasis in Strategic Leadership from Amberton University located in Garland, Texas.

Table of Contents

Table of Figures

Executive Summary

In this eBook, we will examine Design for Test (DFT) guidelines specific to the design of boards to be tested through the boundary scan (BS) registers of IEEE 1149.1-compliant devices. Implementing boundary scan DFT guidelines adds the unique capability of accessing onboard test resources for a non-intrusive test which provides open and short faults coverage. Following DFT guidelines during the board design process makes the board easier to test for defects. DFT guidelines provides an opportunity to increase test coverage of the board. As boards move through the manufacturing process, the goal is to reduce the number of defective boards produced. Defects need to be identified quickly so they can be repaired, and processes can be adjusted. Boundary scan can also be used to test memory devices and for configuration of programmable logic devices (PLD) and flash memory devices. The DFT guidelines contained herein have been assembled over many years of experience by the technical staff of ASSET InterTech, Inc. and validated across a variety of simple and complex board designs.

Board Level Design for Test (DFT) Guidelines

Figure 1: Board Level Design for Test (DFT) Guidelines

The first guideline in this eBook is the most important. As a board designer, where you have a choice, maximize the use of 1149.1-compliant devices. The more boundary scan register access you have, the more fault-coverage can be realized, between boundary scan and boundary scan devices and between boundary scan and non-boundary scan devices. Higher fault coverage can be achieved by using unused boundary scan pins to control buffer direction and output enable signals of simple devices like Texas Instruments ‘244 and ‘245 buffer devices. Also, using validated BSDL files for the boundary scan devices is suggested.

Board Level Boundary Scan Infrastructure

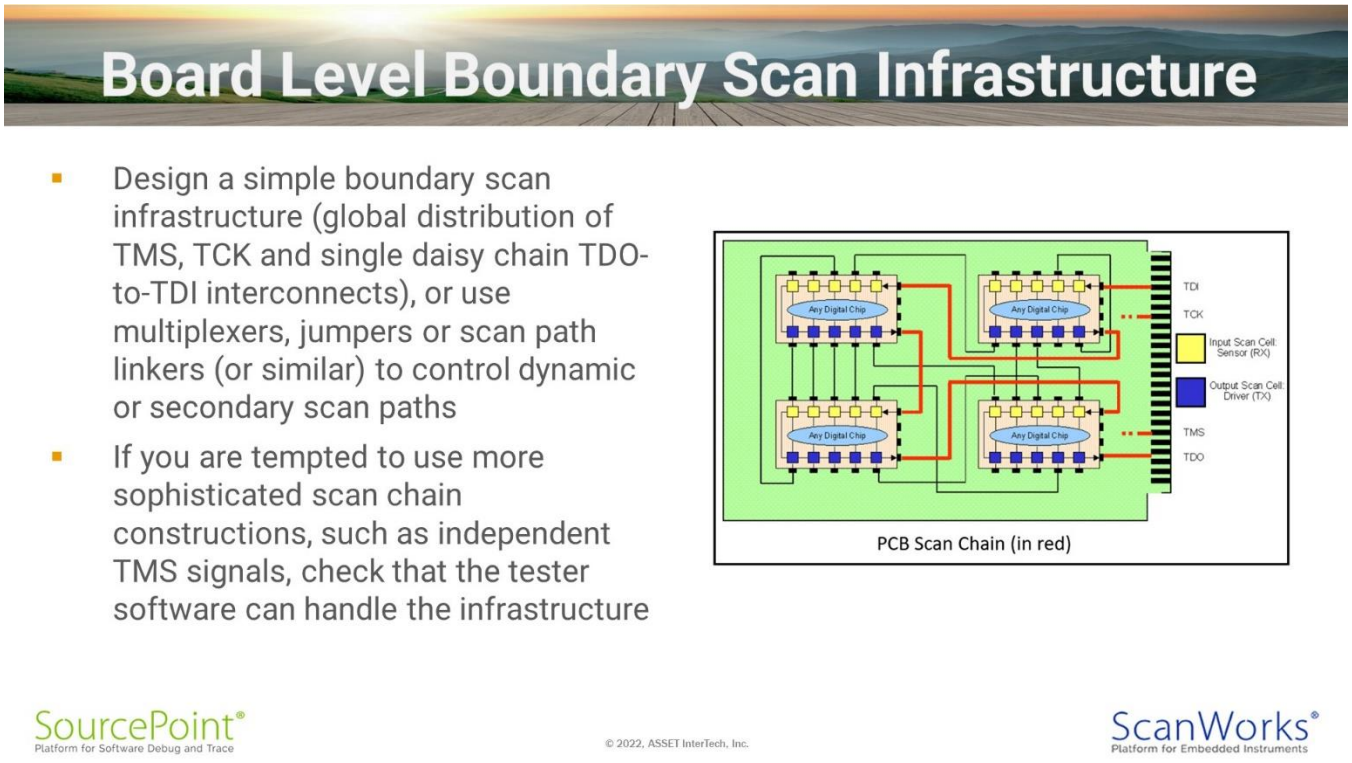


Figure 2: Board Level Boundary Scan Infrastructure

It is possible to design complex scan chain configurations at the board level (e.g., different Test Mode Select (TMS) lines going to subsets of different 1149.1 devices, direct access rather than daisy chained access to individual Test Data In (TDI) and Test Data Out (TDO) pins, etc.). The advice here is to make sure the PC-based board tester can manage such complex scan chain infrastructures.

Access to Test Access Port (TAP) Signals

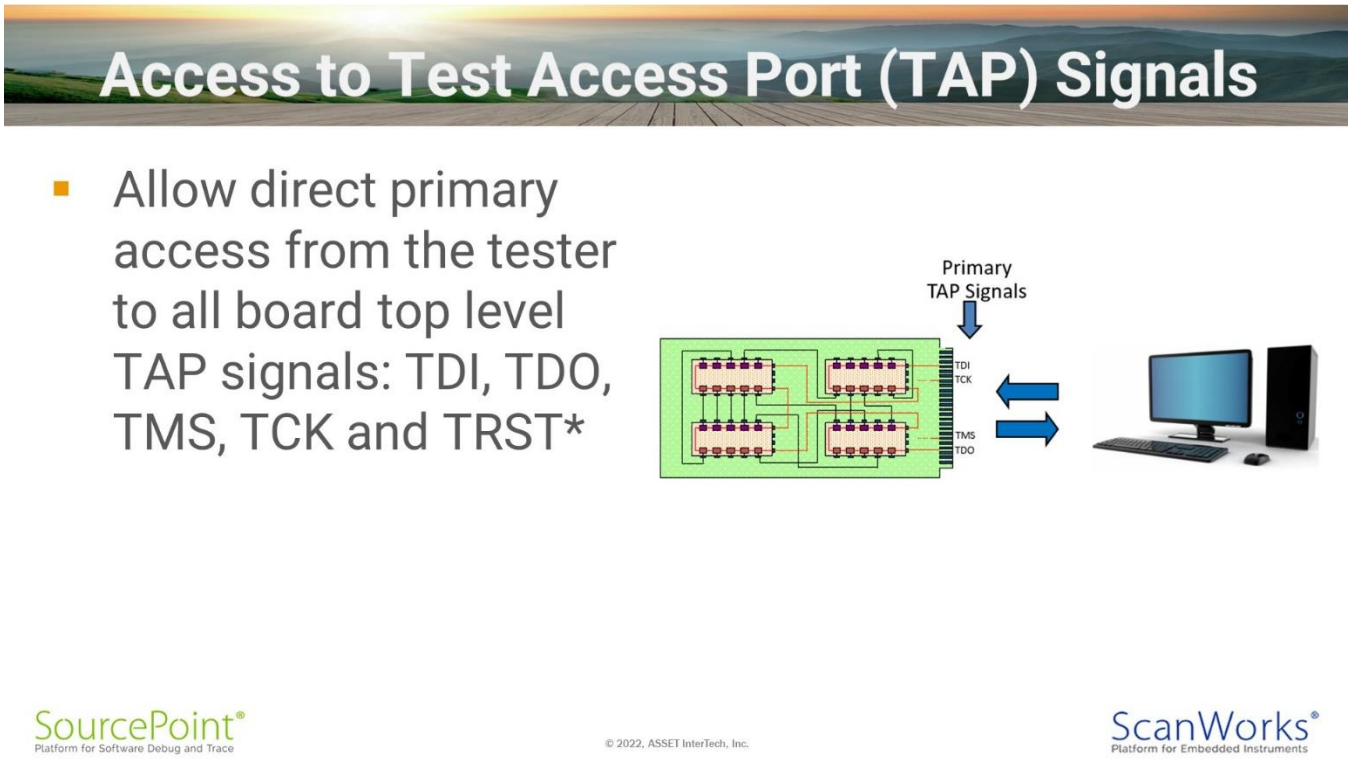


Figure 3: Access to TAP Signals

Be careful with the design and distribution of the onboard Test Access Port (TAP) signals. Allow direct access to all TAP signals from the primary contact to the board, edge connector, or plug and socket. Treat both Test Clock (TCK) and (TMS) as critical signals (i.e., properly balanced, no skew, properly buffered with no inversion). Terminate the signals to avoid reflections. The maximum TCK frequency is determined by the slowest device on the board. Maximum TCK frequency for a device is specified in the BSDL file for that device. Place a weak pull-down on the Test Reset* (TRST*) signal.

Buffering TAP Signals

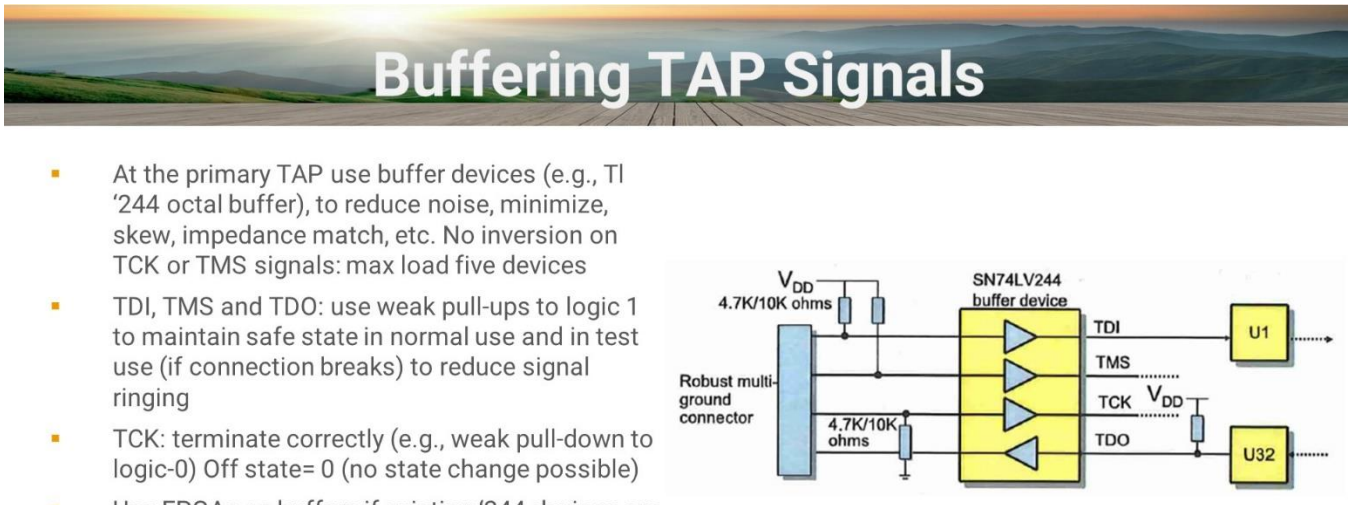


Figure 4: Buffer TAP Signals

Buffering the primary TAP signals on/off the board is recommended to:

- prevent noisy backplane signals from reaching the onboard devices
- handle impedance mismatches between tester and the board, tester drivers and board fanouts, and compensate for cable length
- allow for a faster speed TCK because impedance mismatch issues can be minimized and matched with the tester through a special buffer board
- not limit the cable length between the tester and the board
- allow safe states to be maintained on the signals during normal use and during test use (if the tester-to-board cable becomes disconnected)

Where designs have buffers and you want to change them for scan devices, the usual problem is that scanable buffers tend to be slow or the wrong voltage/technology. Therefore, programmable logic devices can be implemented. It is easy to design a CPLD or FPGA to act as a buffer of almost any

complexity. You may even absorb several buffers into one programmable logic device, saving board space. Programmable logic devices are usually fast enough to be utilized on the design, inexpensive, and easy to procure.

Layout of TAP Signals: TCK and TMS

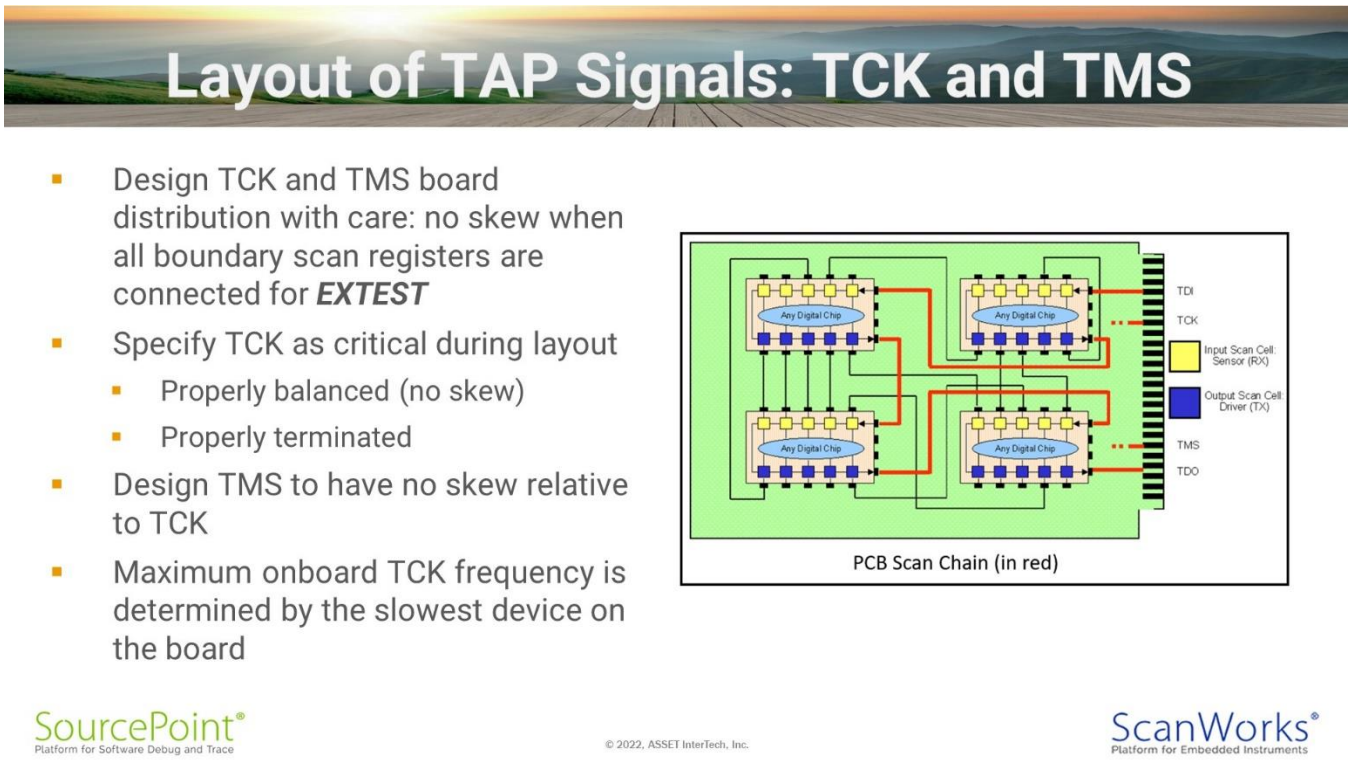


Figure 5: Layout of TAP Signals: TCK and TMS

Be careful with the design and distribution of the onboard TAP signals. Treat both TCK and TMS as critical signals (i.e., properly balanced, no skew, properly buffered with no inversion). However, what is important is that TMS is stable when a value change on TCK occurs. Terminate the signals to avoid reflections. The TCK frequency is determined by the slowest device on the board. Maximum TCK frequency for an 1149.1 device is specified in the BSDL file for that device.

Layout of TAP Signals: TDI and TDO

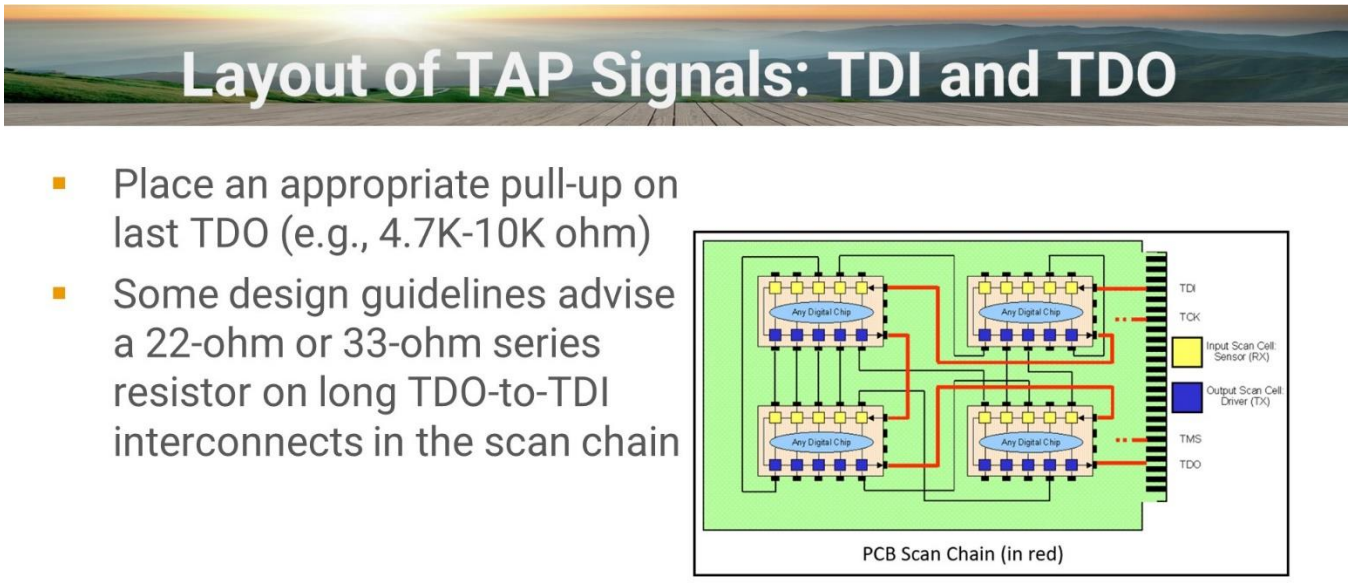


Figure 6: Layout of TAP Signals: TDI and TDO

TDI, TDO and TRST* signals also require attention in terms of drive capability, series resistors, terminating resistors and pull-up and pull-down resistors. The values quoted in the slide are typical values found in the literature on this subject.

Board TRST*: Floating or Tied?

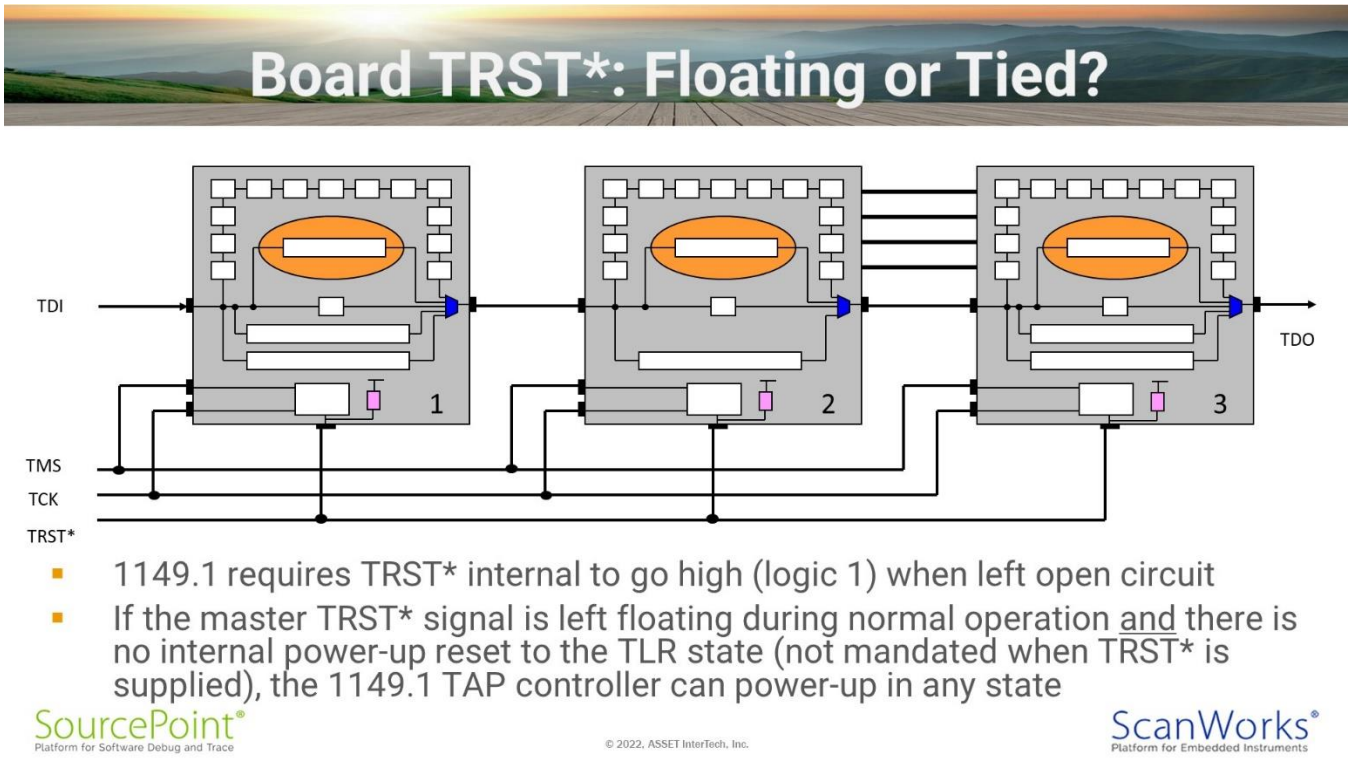


Figure 7: Board TRST*: Floating or Tied?

Section 6.3 of the 1149.1 standard states that, "The TAP controller shall be forced into the Test-Logic-Reset controller state at power-up either by use of the TRST* signal or as a result of circuitry built into the test logic". It is not necessary to provide an internal power-up reset if the TRST* signal has been supplied. If this is the case, what should a designer do with TRST* lines on a board once the board has been tested and is now working normally in a system, leave it floating or tie it low?

The 1149.1 standard requires TRST* to go high (logic 1) when left open circuit so that the device is tolerant to an open circuit fault on the TRST* pin of any 1149.1 device. However, this means that if the master TRST* signal is left floating during normal operation, the 1149.1 devices can potentially power-up in an unsafe state. One solution is to hold TRST* constantly low while the board is operating normally. Additional solutions are discussed in the following slides. If the TRST* signal is not incorporated into the 1149.1 devices, the device must contain an internal power-up reset.

One Potential Consequence of No POR

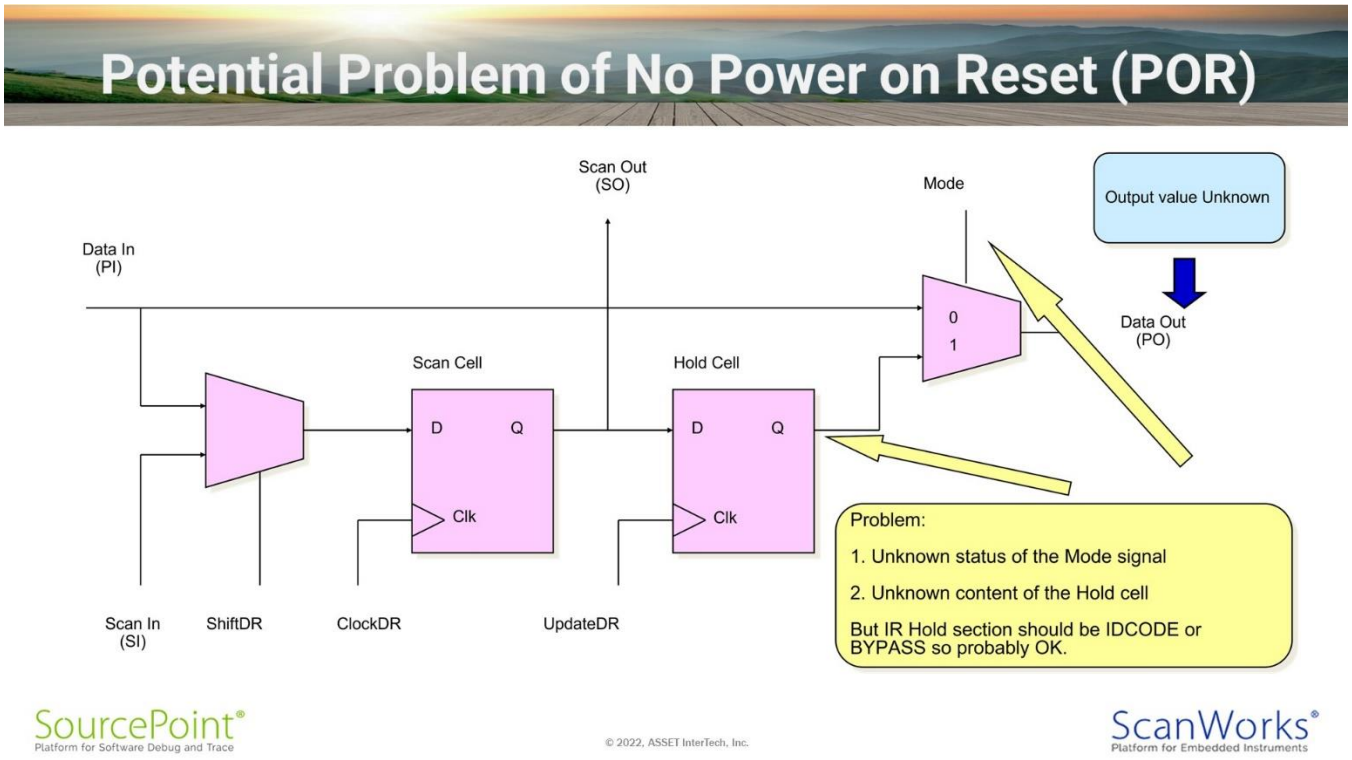


Figure 8: Potential Problem of No Power on Reset (POR)

Here, the potential problem of no Power-on Reset (POR) is illustrated with the BC_1 boundary scan cell. The output multiplexer Mode control signal is generated by the decoded output of the Instruction register. If the register is not initialized (to the Test-Logic-Reset state), then the value of the Mode control signal could be unknown. If the Mode control line powers up as logic 1, then the unknown state of the Hold Cell will be passed to the output pin.

The Hold section of the Instruction Register should power up with the opcode for the IDCODE instruction (if the Identification register is present), else for the BYPASS instruction. Both these instructions maintain the boundary scan device in its functional mode, not in test mode. In functional mode, the Mode signal on a BC_1 boundary scan cell should be logic 0, not logic 1. So, although the state of the boundary scan cell may be unknown, this unknown value should not be presented to the output pin.

Power-up Reset on Board

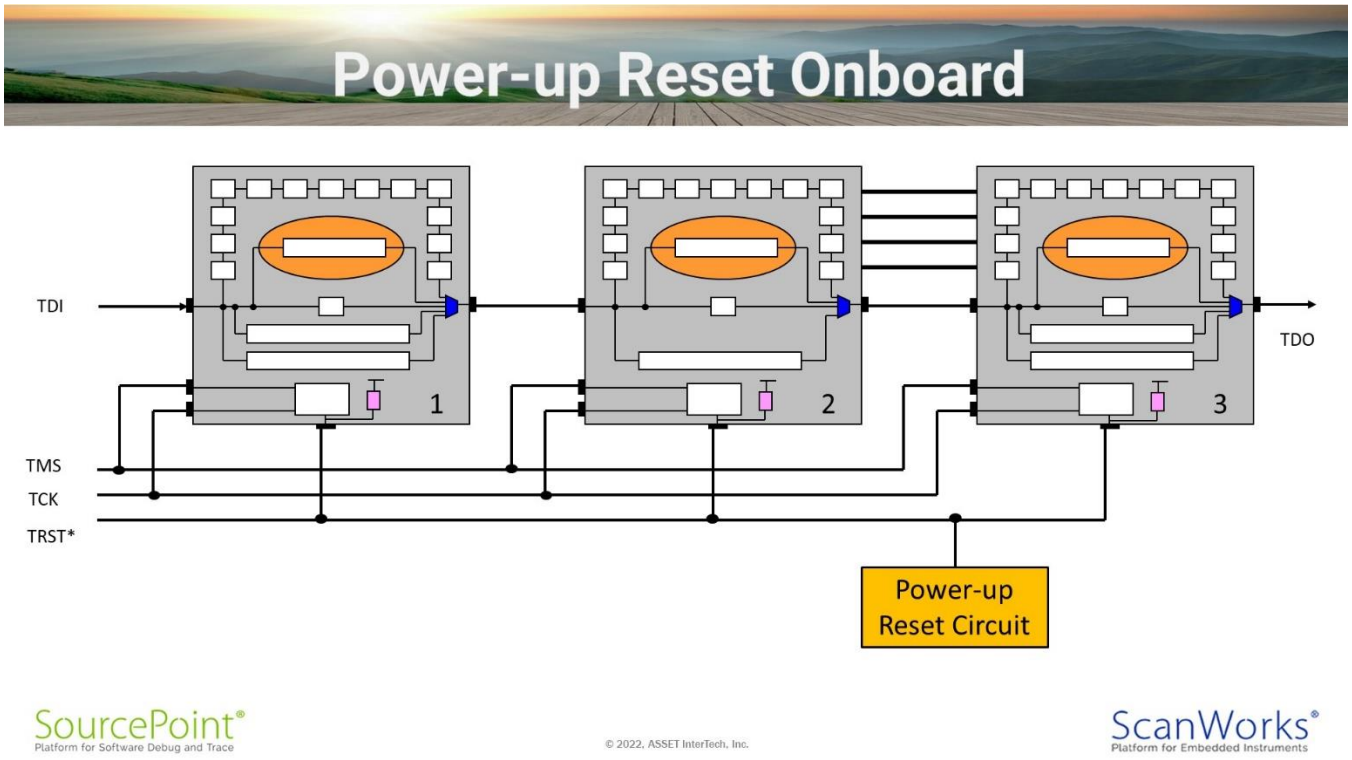


Figure 9: Power-Up Reset Onboard

One solution is an automatic power-up reset circuit on the board. This solution is simple but usually not implemented. The next slide shows the most common solution.

Board Pull-Down Resistance

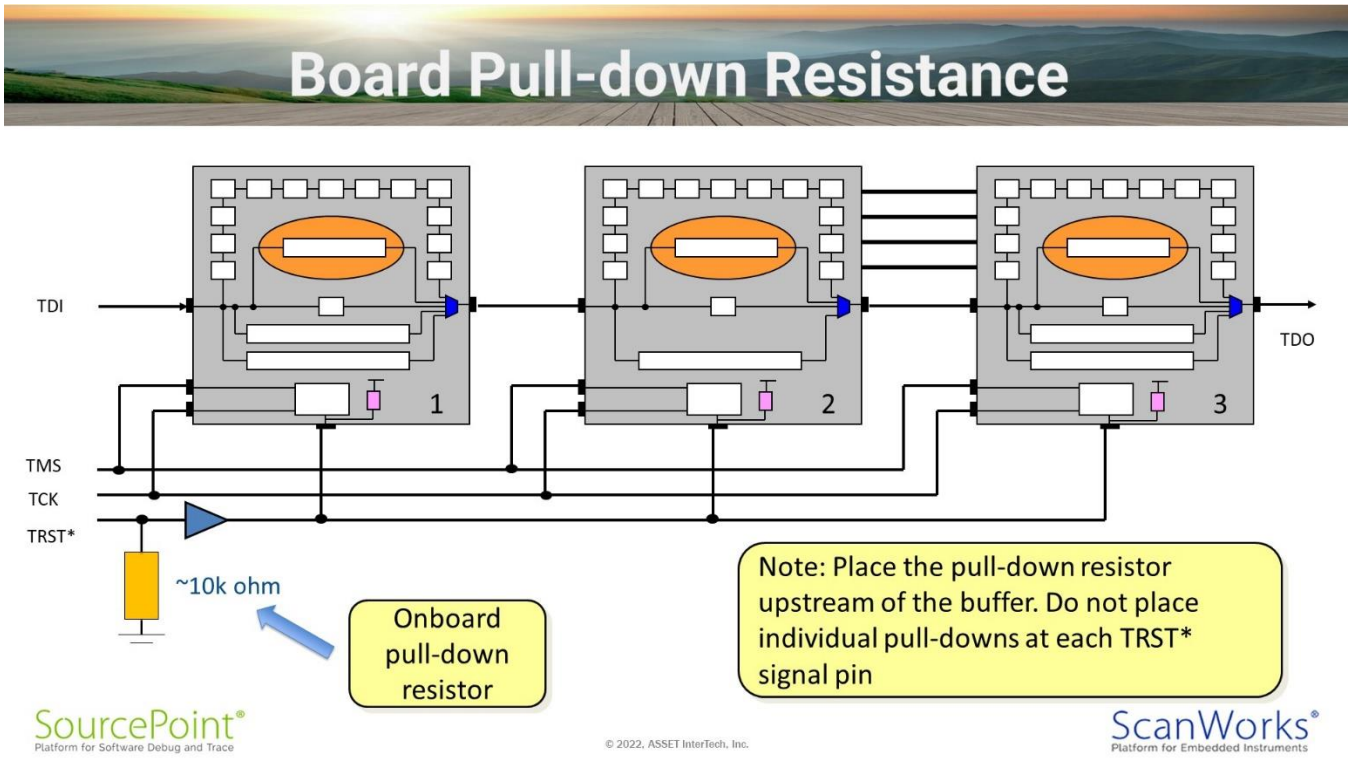


Figure 10: Board Pull-down Resistance

Another solution is to place a weak pull-down (typically around 10K ohm) on the TRST* line that:

- overcomes the parallel sum of the device internal pull-ups
- easily driven high by a PC-based board tester

This solution is common, but the value of the pull-down resistor is critical if there is no buffer on the TRST* signal input. If the pull-down resistor is too low, a PC-based driver may not be able to drive TRST* high enough to be sensed reliably as a logic 1 by all boundary scan TRST* inputs. If the pull-down resistor is too high, and the board contains a voltage divider, that may drain considerable power from, say, a battery power supply system.

When the board contains many boundary scan devices with the TRST* signal provided, the parallel sum of the internal TRST* pull-ups will become very low. Under these conditions, it is recommended to use a buffer to drive the primary TRST* signal onto the multiple TRST* signals and then to place the pull-down resistor on the primary side of the buffer, as shown in the figure above.

Internal POR Circuit

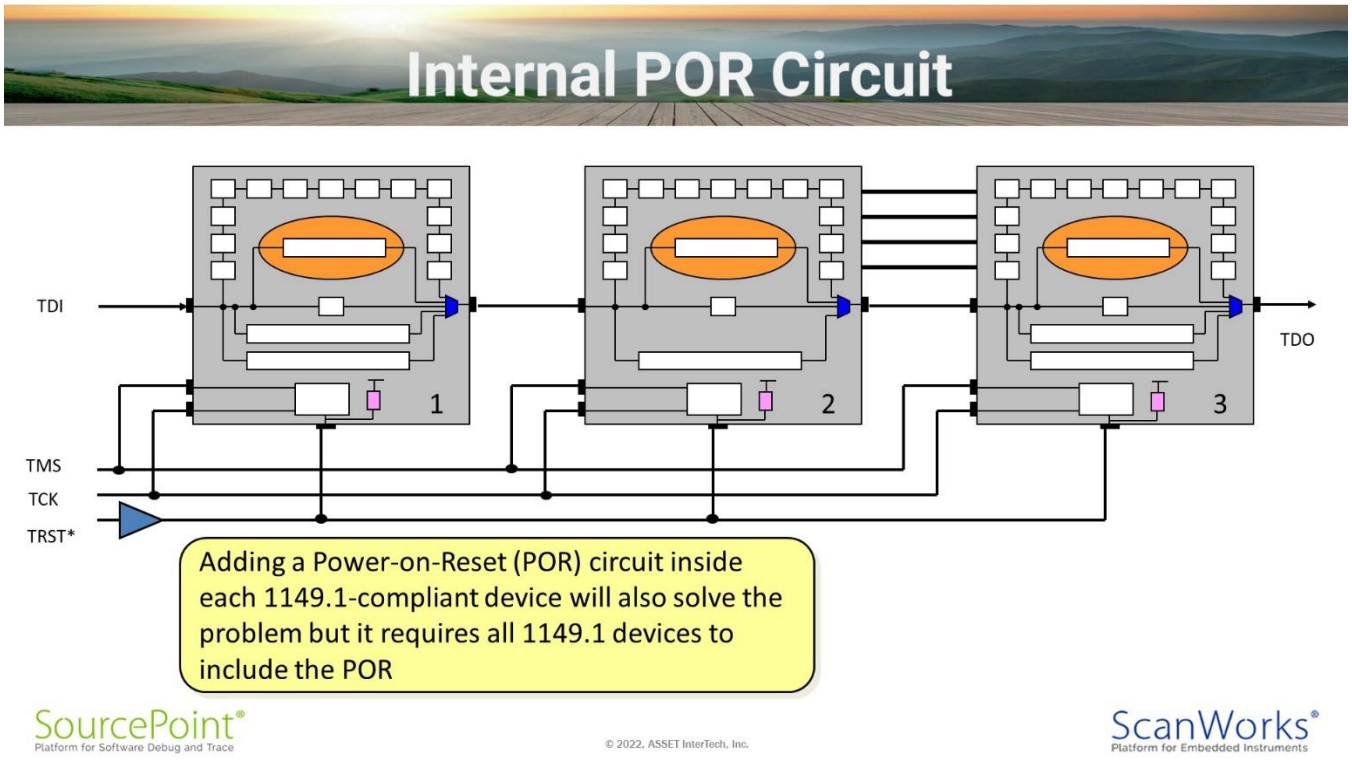


Figure 11: Internal POR Circuit

If each 1149.1 device contains a POR (to the Test-Logic-Reset state), then the problem goes away. However, every device in the chain, including those devices with a TRST* signal, must have this facility to work correctly at the board level.

Testing the Tester: Resolving Opens

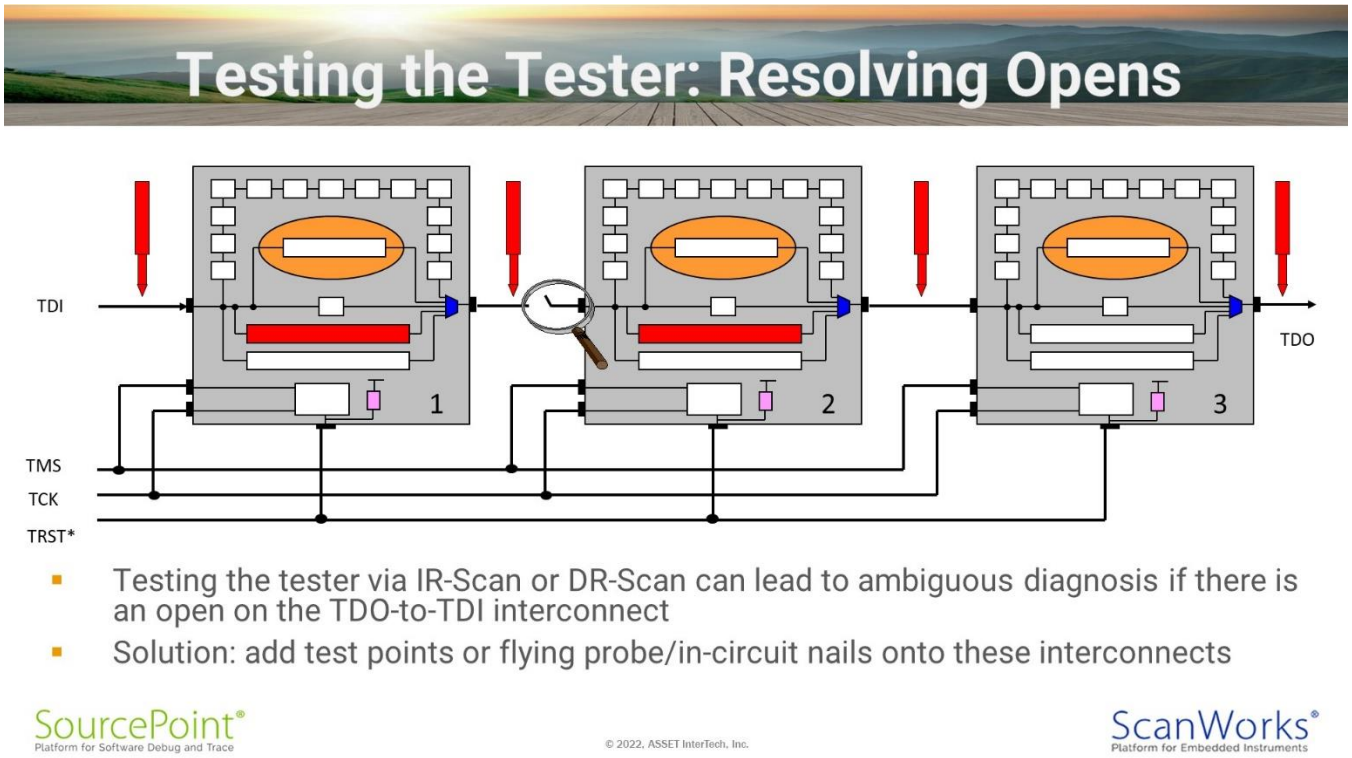


Figure 12: Testing the Tester: Resolving Opens

Testing the tester is accomplished either by an IR-Scan or a DR-Scan. IR-Scan places the Instruction register between TDI and TDO and then captures and shifts out the internal hardwired 01 in the two least significant bits of the register. DR-Scan places the Identification register, if present, else Bypass register between TDI and TDO and then captures and shifts out the internal hardwired values (1 + 31 bits for Identification, 0 for Bypass). If there is an open circuit between a TDO-out and the next TDI-in, it is not possible to identify at which end the open has occurred. Consequently, the diagnosis is to both devices. Placing a test point or flying probe/in-circuit nail on these interconnects is a simple solution that allows the open to be located to just one of the devices.

Tying off Control Pins

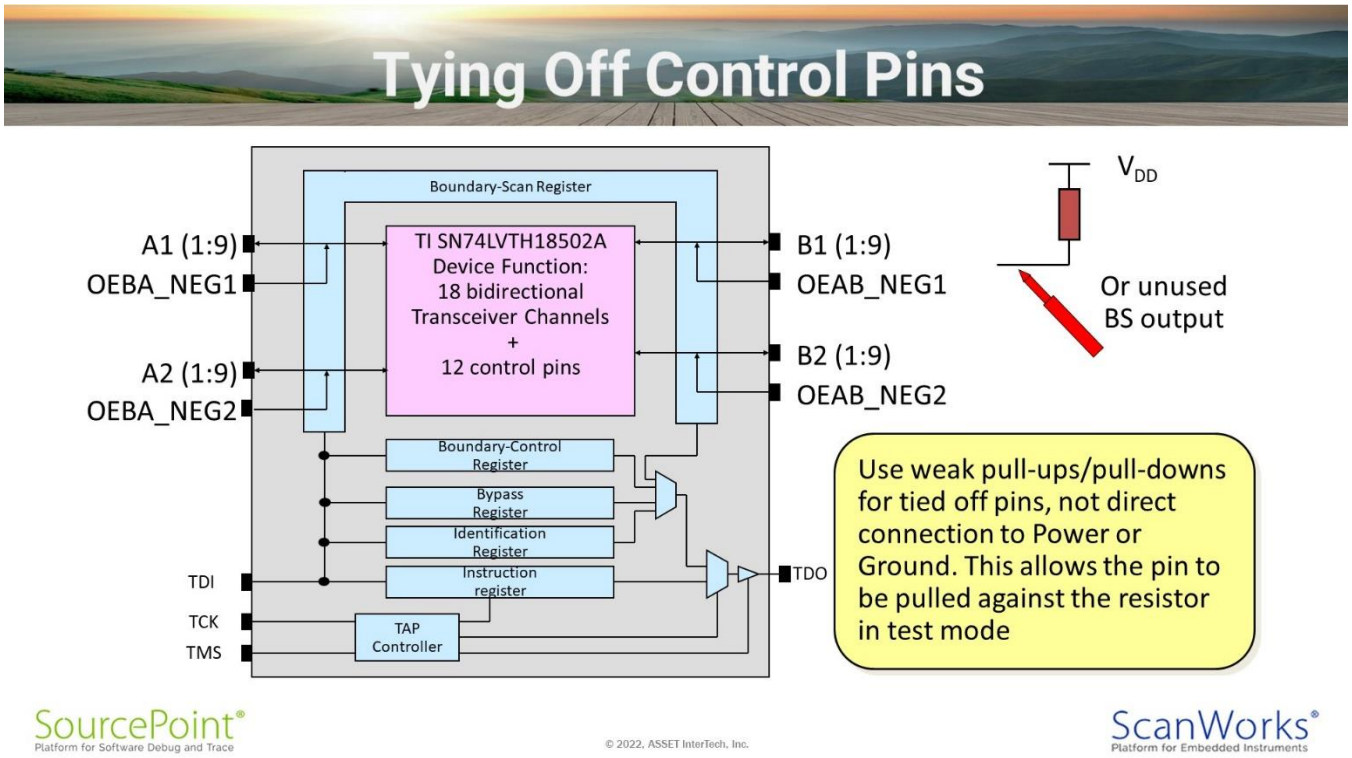


Figure 13: Tying Off Control Pins

If bidirectional (IO) or three-state (OZ) control pins are tied off at the board level, tie off with a weak pull-up or pull-down resistor rather than connect direct to the Power or Ground rails. This allows an in-circuit test (ICT)/flying probe test (FPT) nail to be used during test mode to change the status of the pins, plus it allows detection of a missing or open circuit resistor through a simple parametric test.

Board Level Ground Bounce

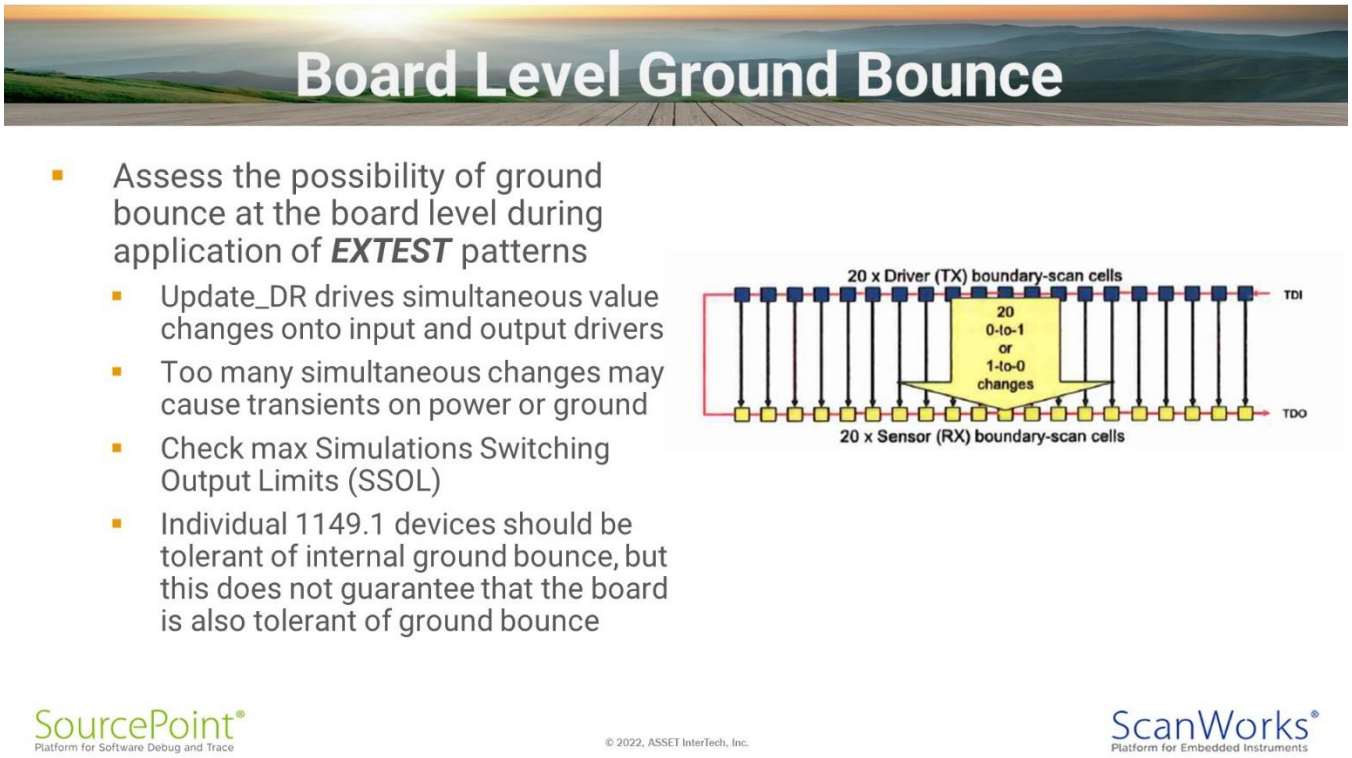


Figure 14: Board Level Ground Bounce

Although each 1149.1 device should not exhibit internal ground bounce under worst case switching conditions, there is no guarantee that ground bounce will not happen at the board level when all devices are switching worst case. Some PC-based testers can accept a specification for a maximum Simultaneous Switching Output Limit (SSOL) and to constrain the interconnect pattern generator to conform to this limit but note that this is not a boundary scan issue. It is more a general board design issue, and the responsibility is with the board designer to ensure no ground bounce problems for the board no matter what operational mode the board is in, functional or test.

Most large FPGAs state in their data sheet that they cannot support switching of more than a specified number of cells at once. Some FPGAs have many boundary scan cells (over 3,000). Supporting switching this number of boundary scan cells at once is not practical.

Achieving SSOL Constraints

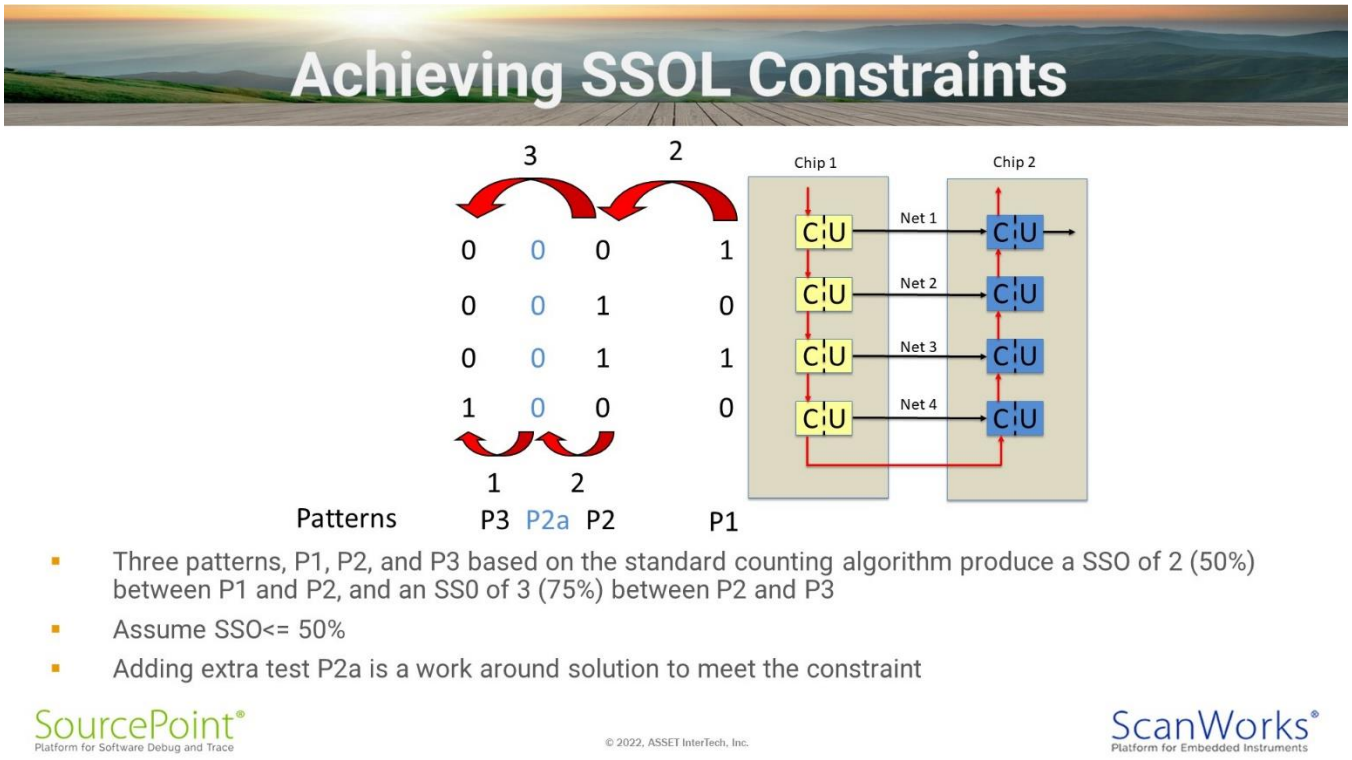


Figure 15: Achieving SSOL Constraints

In the 4-interconnect example above, the patterns have been assigned based on the Counting Algorithm - straight up count. Assume SSOL = 2 (i.e., no more than 50% of the boundary scan cells can switch simultaneously). The Counting Algorithm breaks the 50% constraint limit between Pattern 2 and Pattern 3 if Pattern 3 follows the vertical counting pattern of 0001.

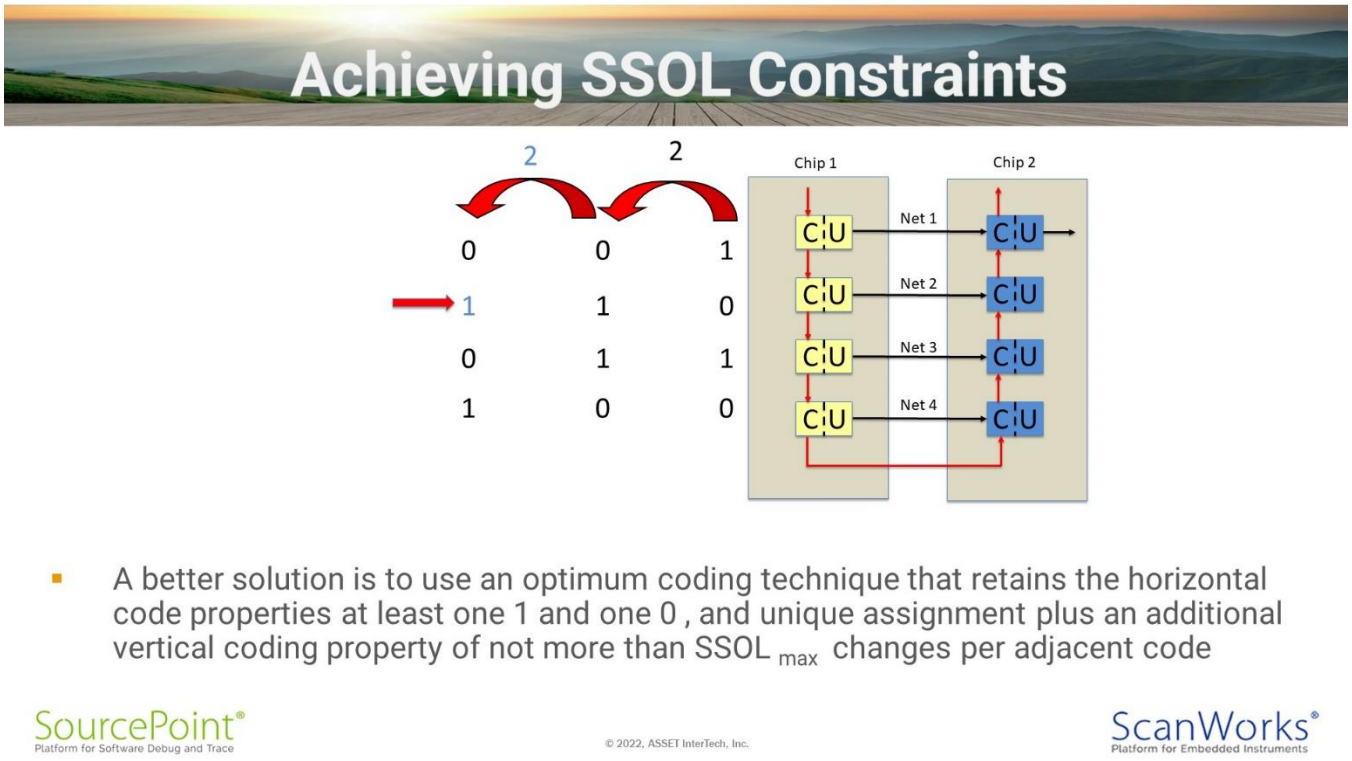


Figure 16: Achieving SSOL Constraints

Extra test, P2a = 0000, is added to reduce SSOs between pattern 2 and Pattern 3 to 2 and 1, respectively. But, changing the horizontal code on Net 2 from 010 to 110 has the same effect and does not incur the extra test. All other basic properties for detection are retained (each code has at least one 1 and one 0, and each code is unique).

BS-to-non-BS Interfaces

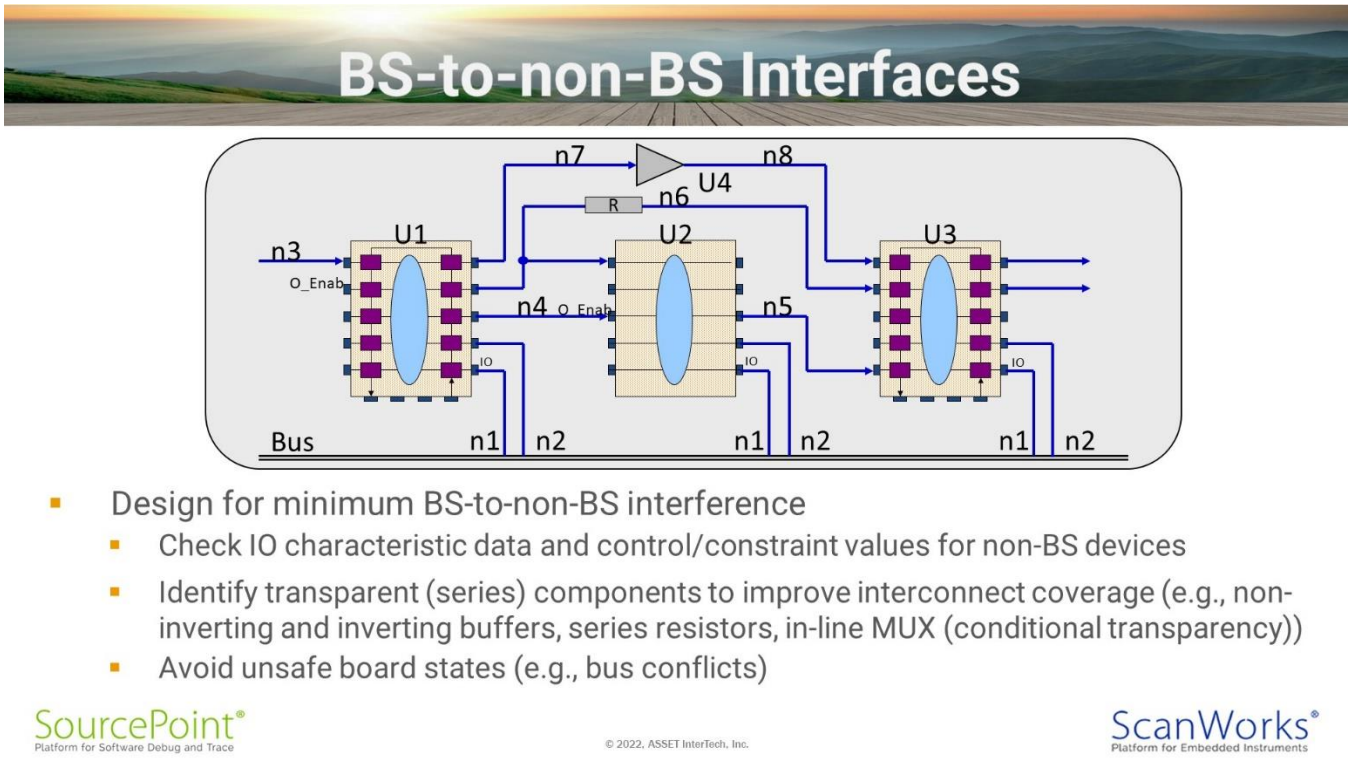


Figure 17: BS-to-non-BS Interfaces

The figure illustrates potential issues of interfacing between boundary scan and non-boundary scan devices and the resulting data needed for the non-boundary scan devices on the board.

The Interconnect pattern generation algorithm will have no awareness of the nature of any other non-boundary scan device attached to boundary scan controlled or observed nets. Therefore, we must supply some basic information about these devices. For example, are the non-boundary scan pins inputs (I), outputs (O), tristate outputs (OZ) or bidirectional (IO)? This data, simple to determine, is called characteristic data or cluster model data. In the example above, we will need to supply characteristic data files (cluster models) for the non-boundary scan devices U2 and U4.

In the case of tristate and bidirectional pins, we need to know what logic values on what pins will control the status of these pins so that, if necessary, we can put them into a safe high-Z or input mode state (no bus conflict). This data is known as constraint data. In the example above, we will need to

constrain the bidirectional pins of U2 to be inputs rather than outputs during interconnect test. This will be achieved by a constraint value on U2's O_Enab signal.

Finally, certain non-boundary scan devices, such as a series resistor or a line driver (U4 above), have a useful property called transparency, that is, logic values on the inputs are passed to the outputs with no change. If the interconnect pattern generator knows about this property, then it may be able to treat two separate nets (n7 and n8 above) as one continuous net and drive test patterns across the single net, thereby increasing the coverage. Make sure all characteristic data for non-BS devices are available, including transparent properties and enabling pins and control value for three-state (OZ) and bidirectional (IO) pins.

Controlling Non-BS Device Output Enables

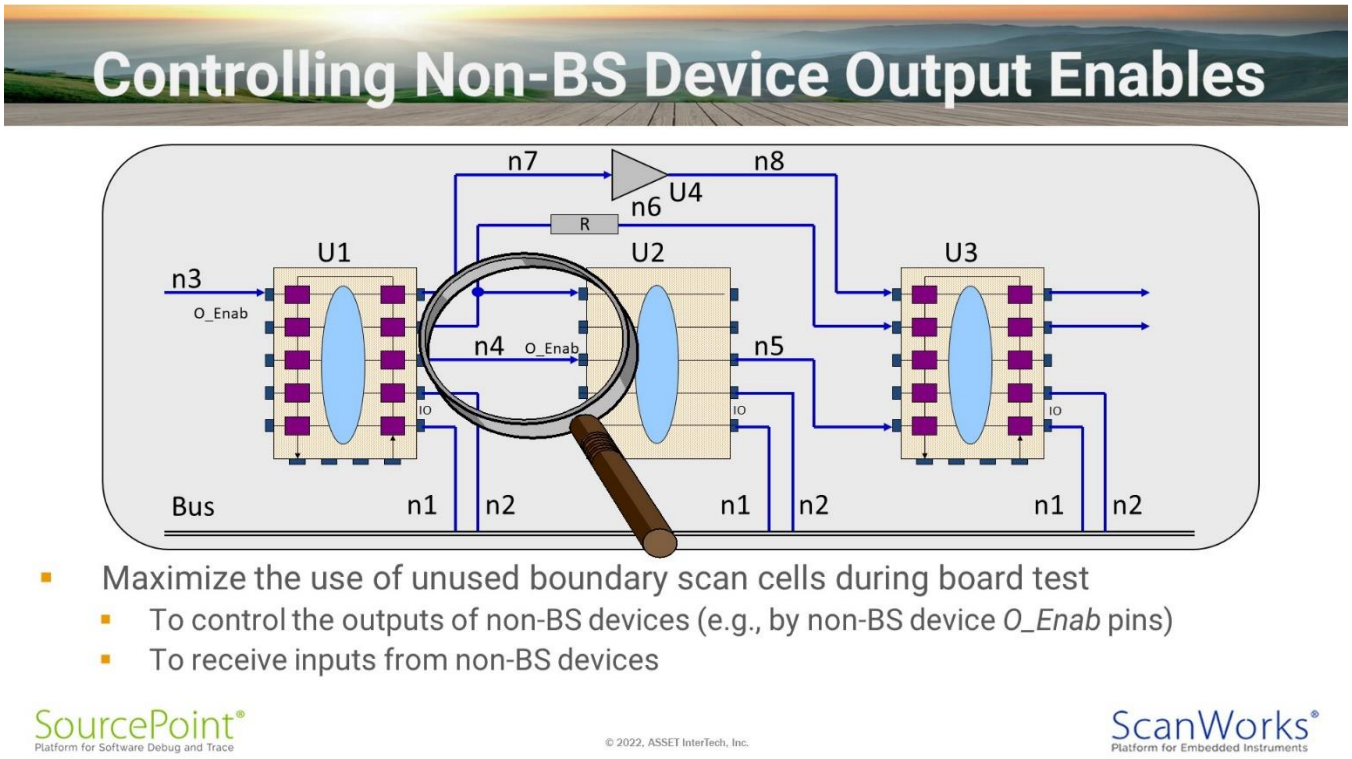


Figure 18: Controlling Non-BS Device Output Enables

A general guideline that comes out of the previous discussion is to make sure that pin status control signals for non-boundary scan devices can be controlled: either directly from the edge connector, by a physical ICT/FPT nail, or by an unused boundary scan cell.

Beware Unsafe Board States

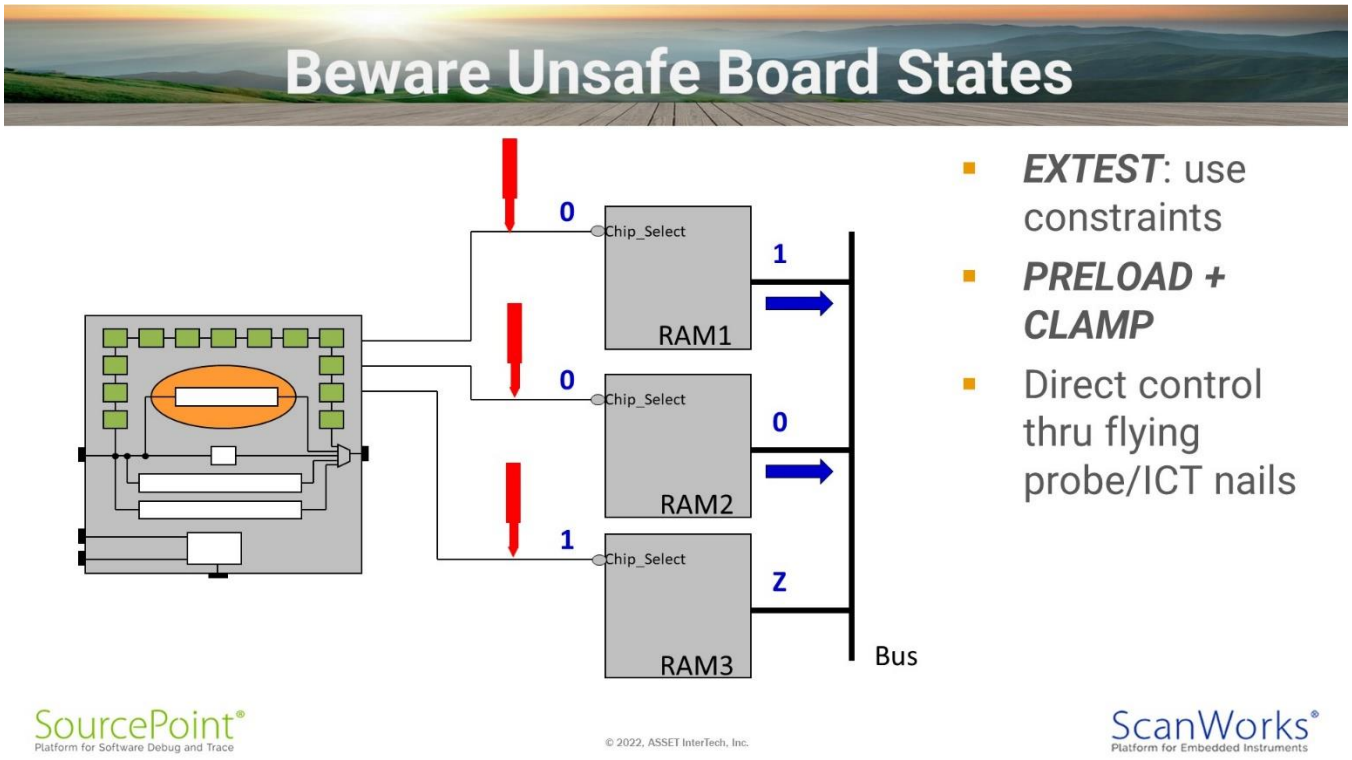


Figure 19: Beware Unsafe Board States

During interconnect test, boundary scan devices are in test mode (EXTEST instruction). If a boundary scan device in test mode happens also to be connected to a non-boundary scan device, the values driven into the non-boundary scan device are coming from the boundary scan cells and are the values created by the interconnect pattern generator. These values have the potential to damage the non-boundary scan device and the board should be designed to prevent this from happening.

The example above shows three non-boundary scan Random Access Memory (RAM) devices whose Chip_Select control inputs are connected to three outputs of a boundary scan device. If the three boundary scan cells are unconstrained, there is a high probability that any two or all three RAM will be selected simultaneously, thus creating a bus conflict.

One solution is to include the hold section in the boundary scan cell (as in the BC_1) and to use the EXTEST instruction to load continuously safe values (known as constraint values) into the hold sections of these three cells. Alternatively, a set of safe values could be set up by the PRELOAD instruction and

applied through the CLAMP instruction, leaving the boundary scan device with its BYPASS register selected. Yet another solution would be to provide an external control on the Chip_Select lines and drive the outputs to their high-Z state.

As an aside, it is worth noting that many BSDL files define X to be the safe state on Tristate and Bidirectional pins. In reality, there is no such thing as an X state and the value will default to either 1 or 0. This value is usually program generation software specific. To avoid surprises, the designer should check that each interpretation of X will not cause damage to driven devices.

Loopback Connector Test

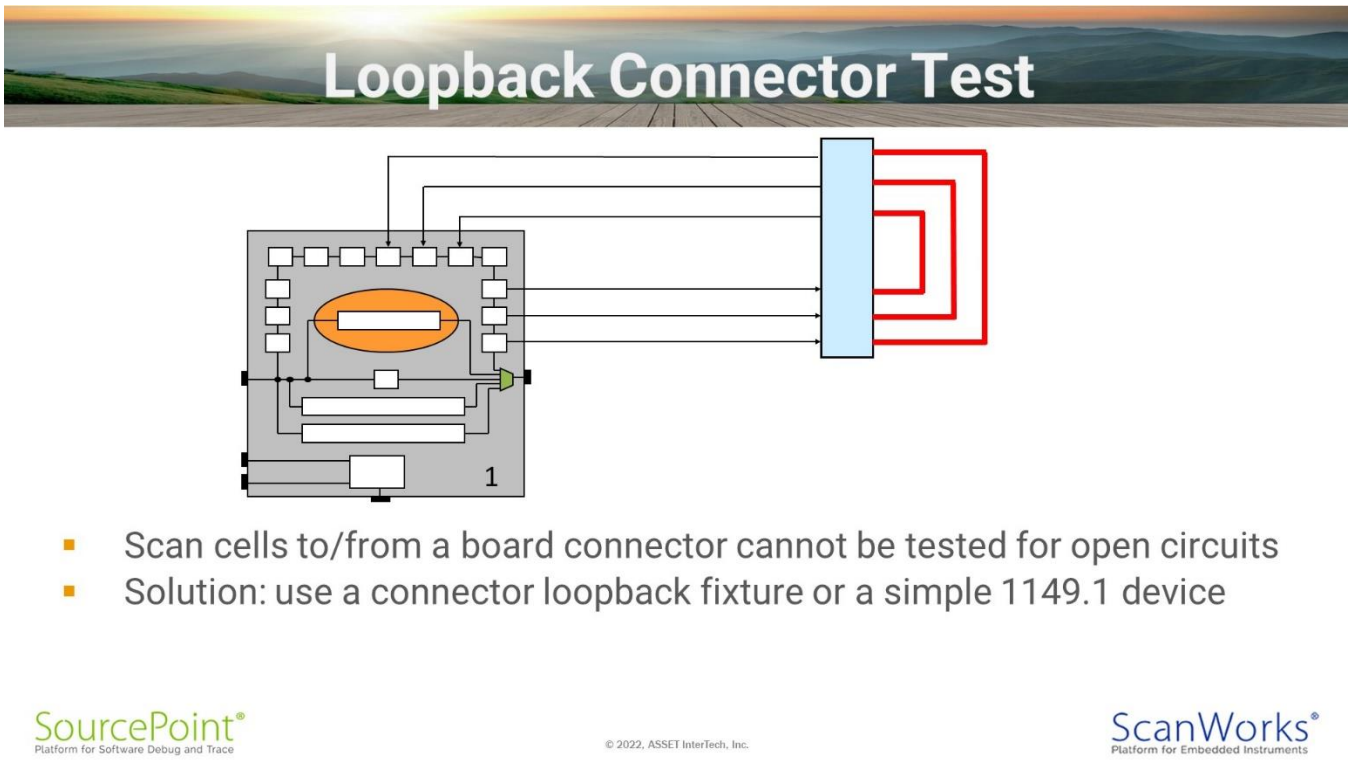


Figure 20: Loopback Connector Test

Are there any boundary scan devices connected directly to an edge connector? If so, can a loopback test be used to check for opens between the scan cells and the connector pins. Alternatively, can you plug into a dummy 1149.1 device for testing purposes.

Different Voltage Levels

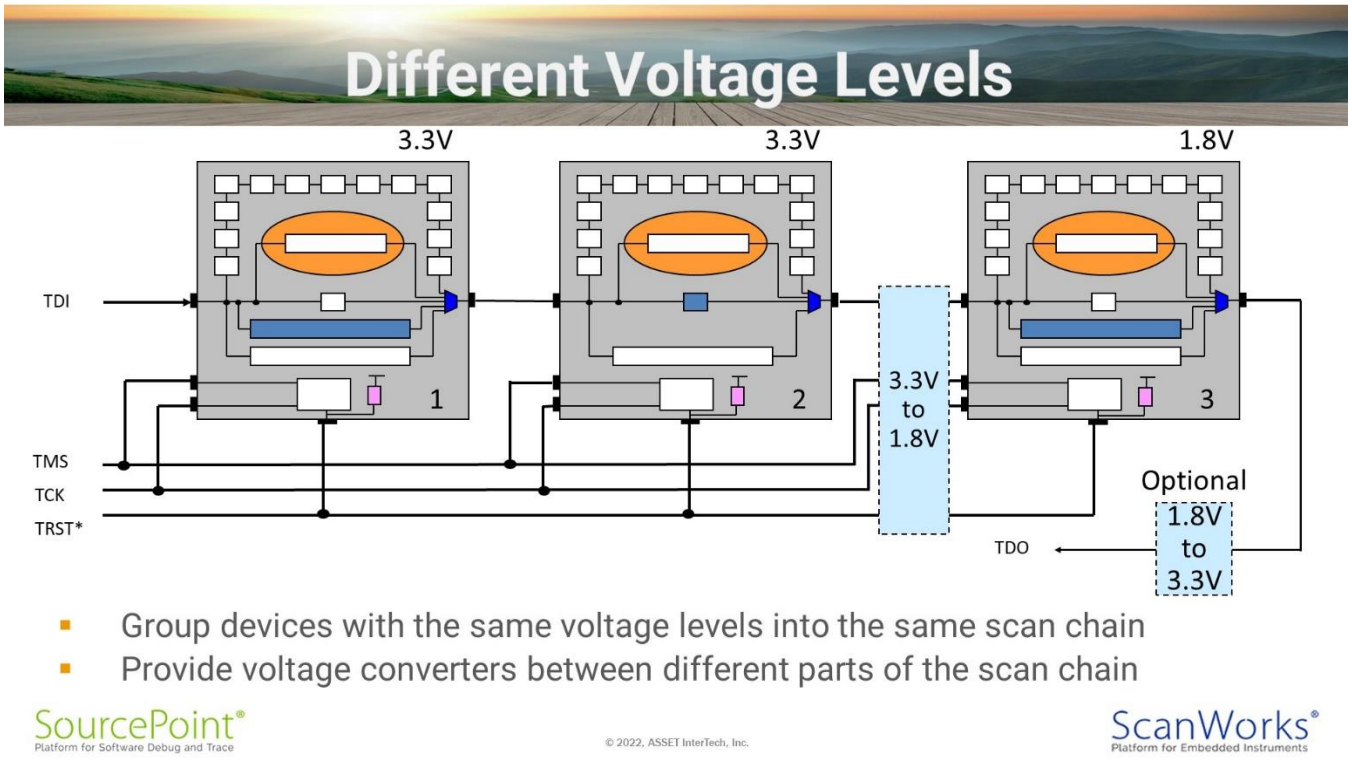


Figure 21: Different Voltage Levels

Devices on the same board with different voltage levels should be grouped and master TAP signals distributed globally, passing through appropriate voltage level converters. Therefore, full interconnect tests can still be applied.

Final TDO off the board may need to be converted back to the same level as the initial TDI, TMS and TCK levels. It depends on the flexibility of the interface pod.

Handling Non-Compliant Devices

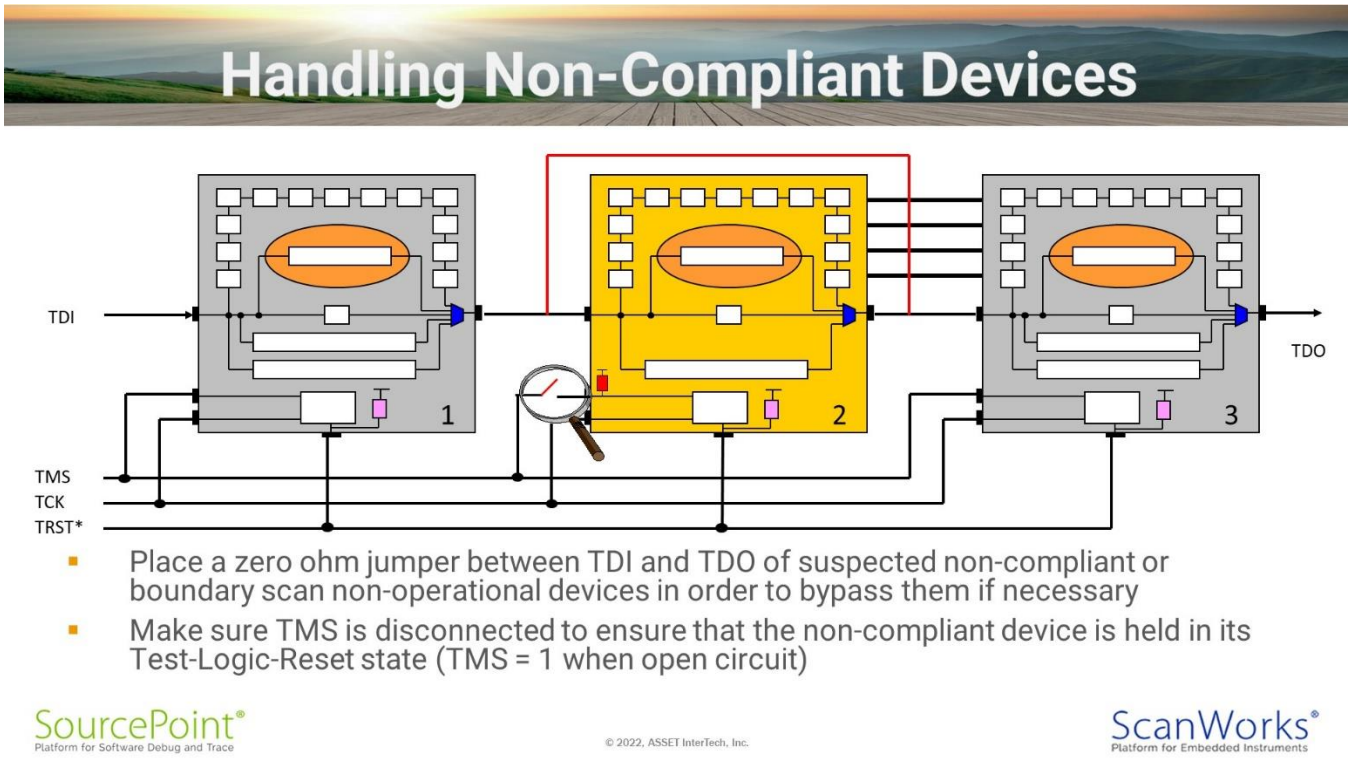


Figure 22: Handling Non-Compliant Devices

Suspected non-compliant devices can cause unpredictable behavior on the board. If the non-compliance is in the TAP Controller, it may be better to switch out TMS and TCK. In this case, one can make a case for a (POR) circuit inside the device even though the device contains a TRST* signal.

In the example above, the non-compliant device is bypassed (direct link from its TDI pin to its TDO pin) and TMS is open circuited to ensure that the device is held in its Test-Logic Reset state (based on the TMS = 1, 5 x TCK synchronizing sequence) and thus non-responsive to any further change on its TDO and TCK pins.

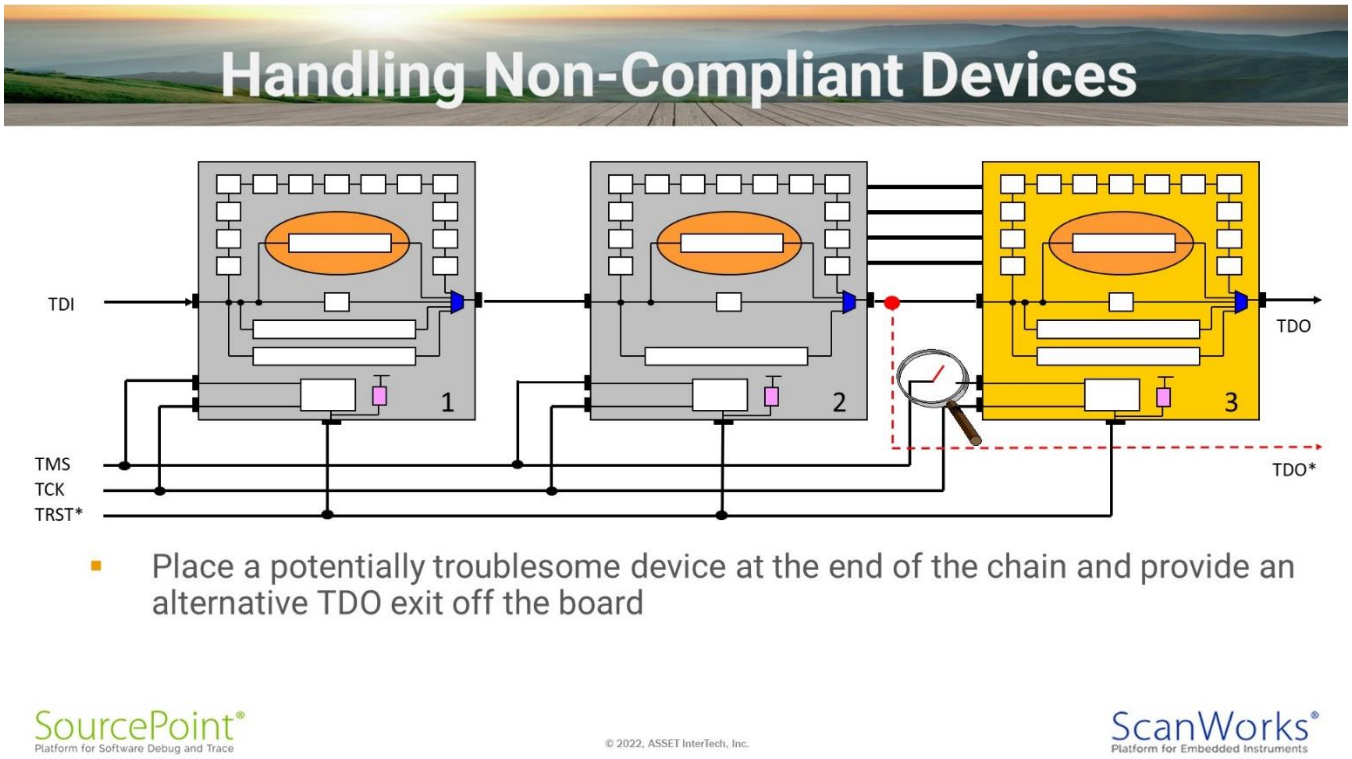


Figure 23: Handling Non-Compliant Devices

Another solution is to place the potentially troublesome device at the end of the chain, allowing it to be bypassed using an extra TDO-exit. See TDO* above.

Using Unused Boundary Scan Cells

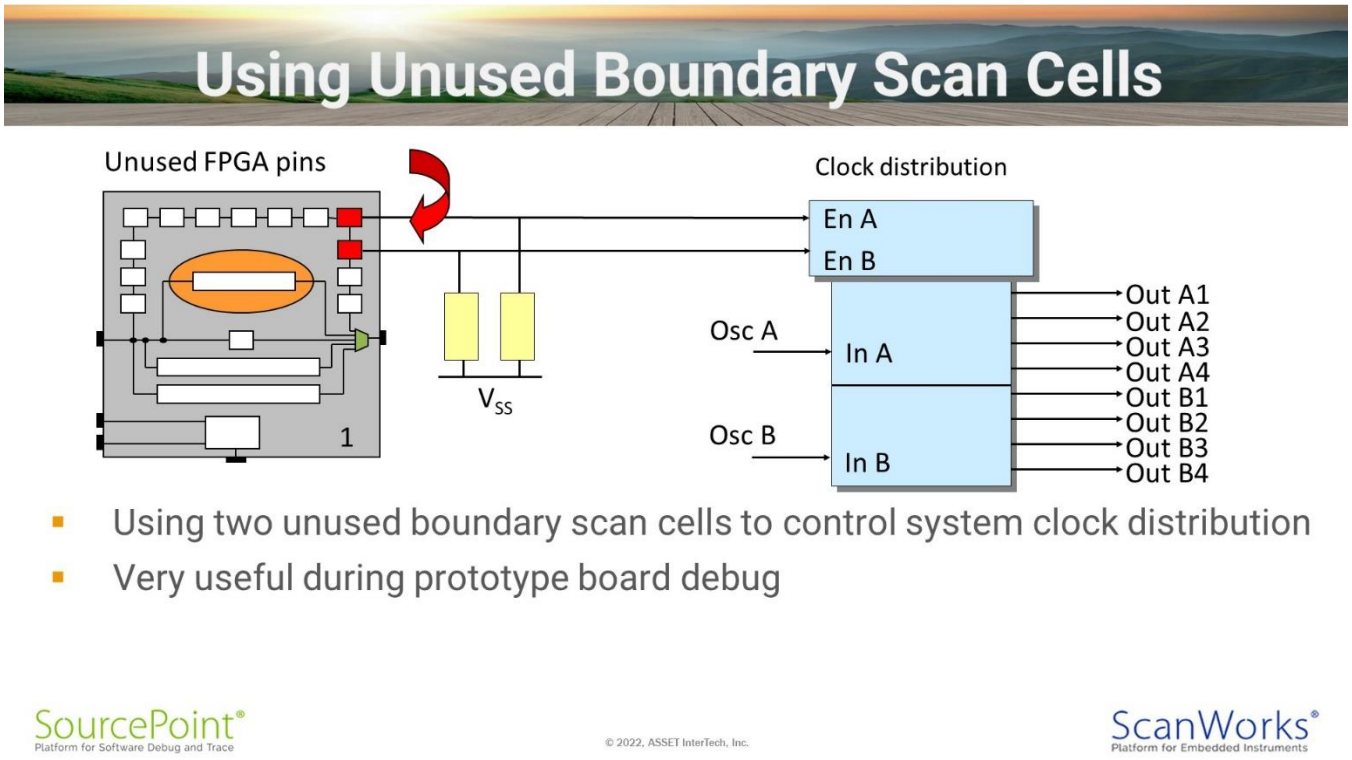


Figure 24: Using Unused Boundary Scan Cells

Are there any unused boundary scan outputs (e.g., from FPGA devices which can be used to increase the controllability or observability of non-boundary scan devices)?

Onboard Clock Generator Control

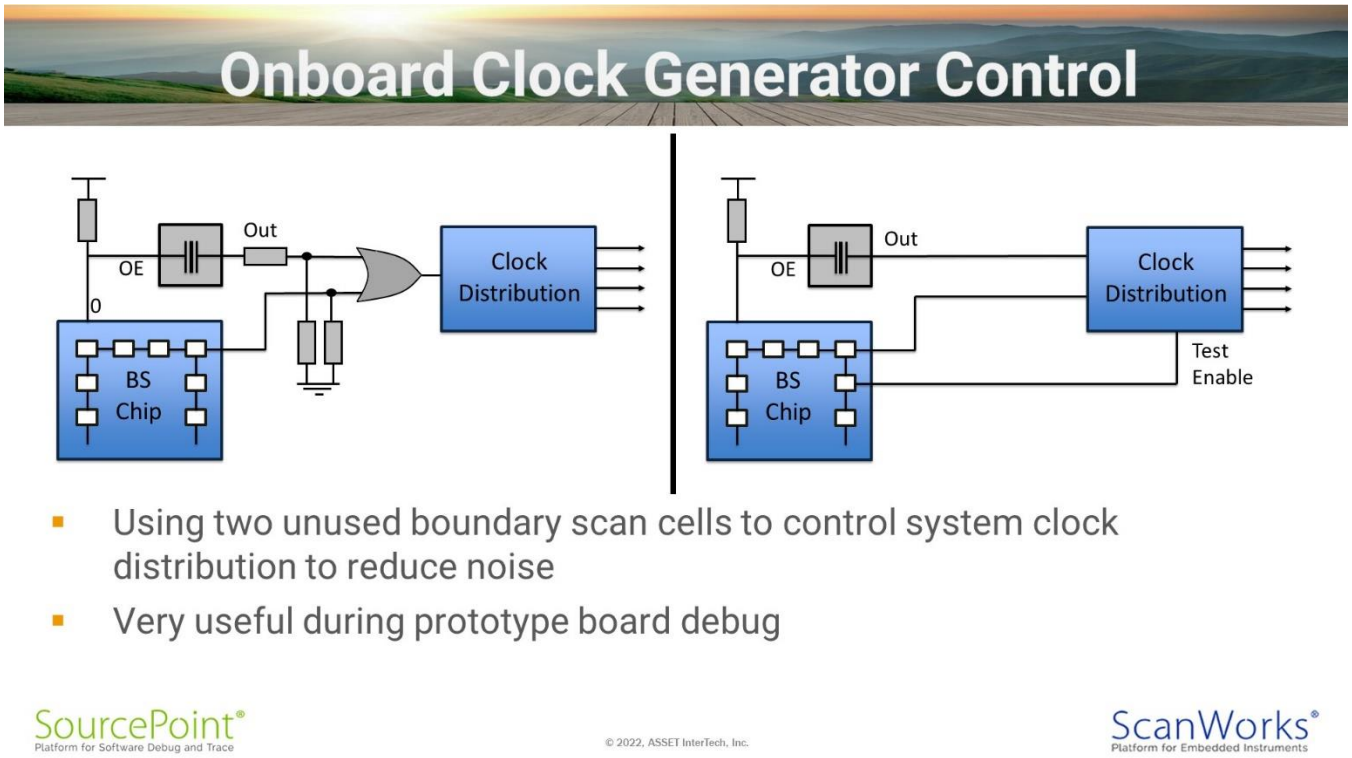


Figure 25: Onboard Clock Generator Control

Free running clock or watchdog timers should be directly or indirectly controllable via boundary scan cells. It should be possible to disable onboard oscillators to quiet the board down and ensure no functional operations take place during boundary scan test mode.

Allow Defeatable Tied-Off Control Pins

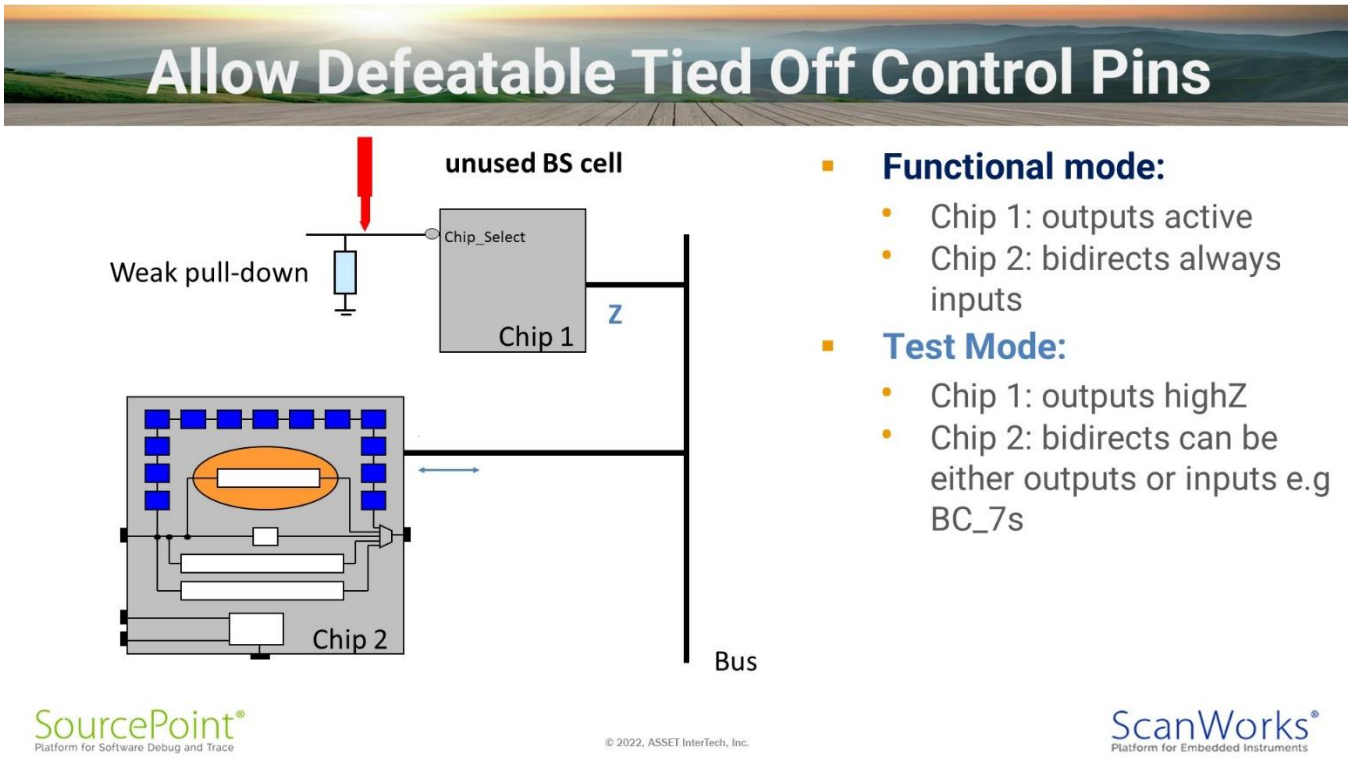


Figure 26: Allow Defeatable Tied Off Control Pins

It may happen that devices are constrained to behave in a certain way during functional mode. In the example above, the 3-state pins of Chip 1 are always used as outputs (Chip_Select signal is set to logic 0) and the bidirectional pins of Chip 2 are always used as inputs during normal functional behavior. In test mode, it might be useful to use the Chip 2 bidirectional pins as output stimulus drivers - maybe to other boundary scan devices on the bus. If this is the case, Chip 1, 3-state output pins should be placed into a safe high-Z mode. If Chip 1 Chip_Select input is tied directly to Ground, the test mode cannot be established. If the Chip_Select signal is tied to Ground through a defeatable pull-down, then a physical nail or unused BS cell can be used to override the weak 0 with a strong 1.

Placing Real Nails: Other Examples

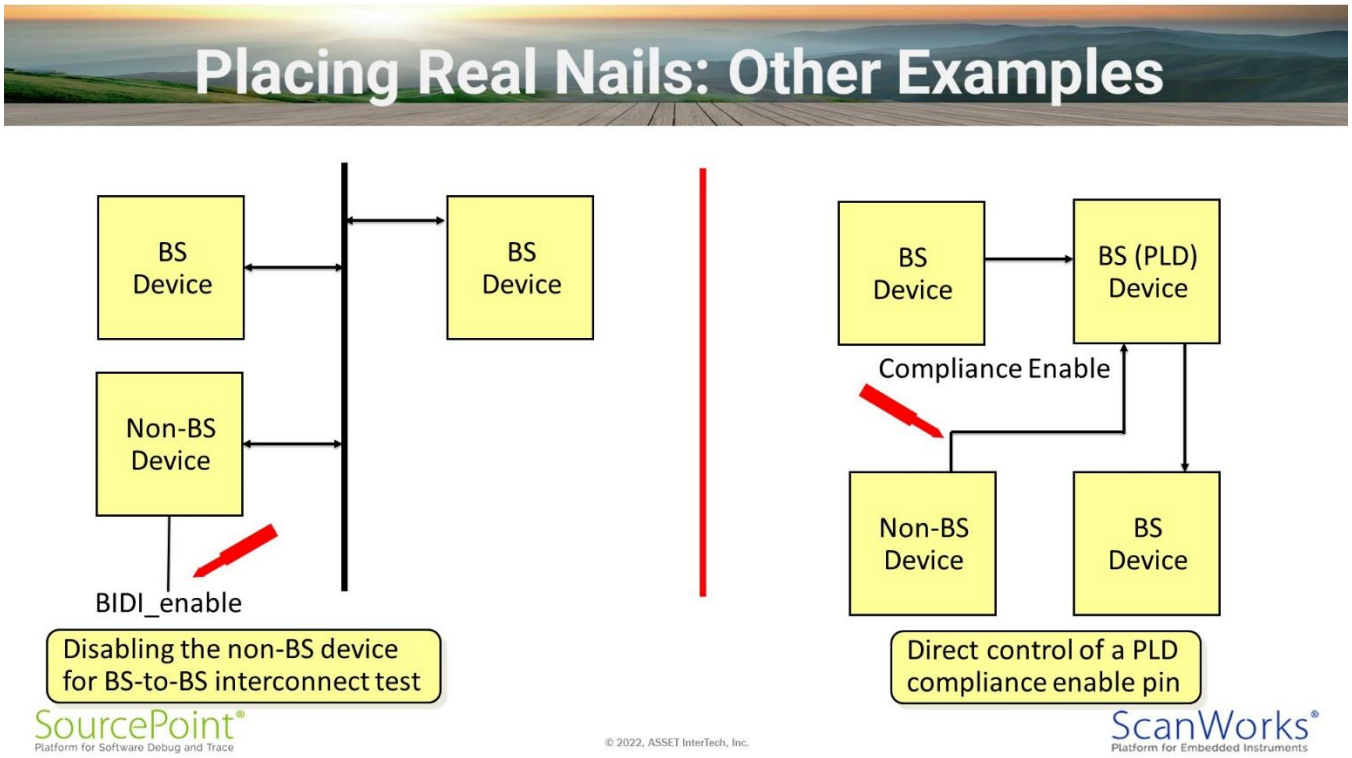


Figure 27: Placing Real Nails: Other Examples

More examples on "good" places to position real nails to assist boundary scan tests.

Use of Physical Nails: Summary

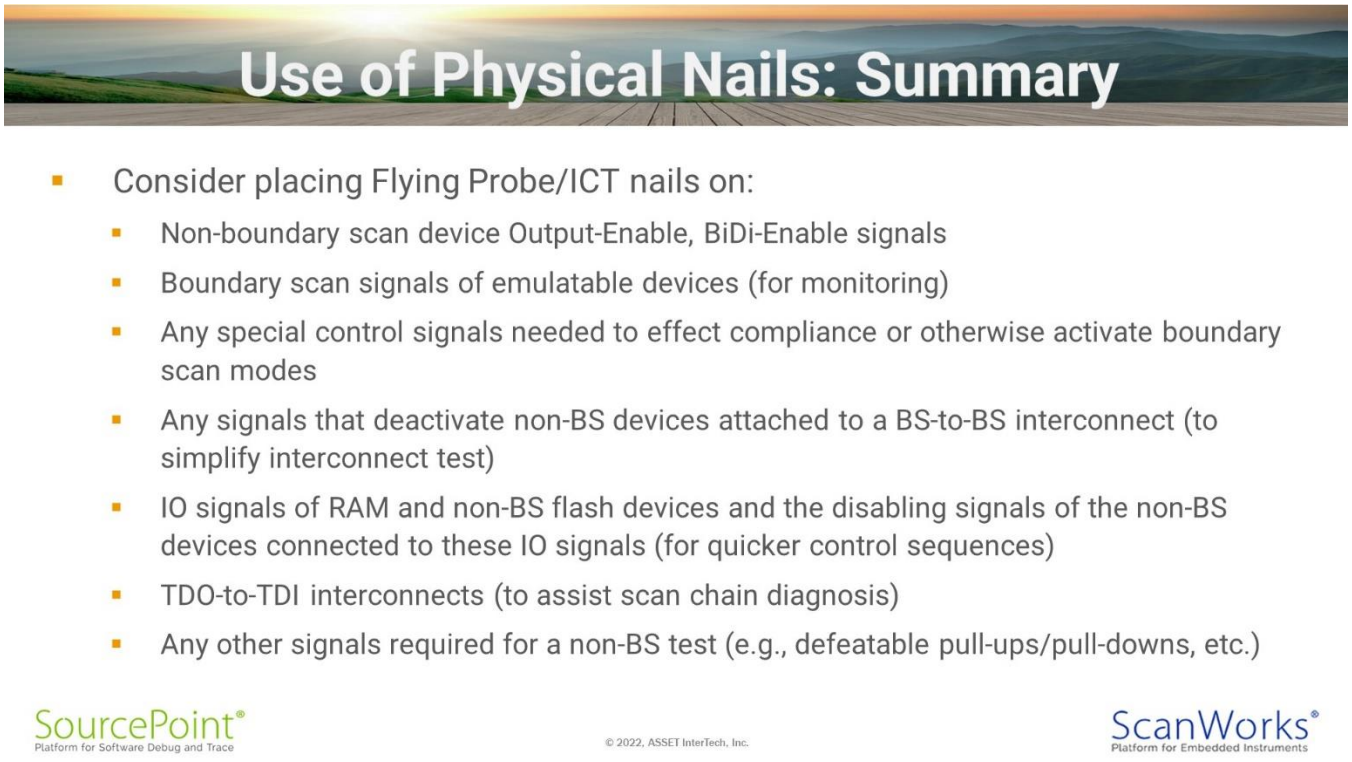


Figure 28: Use of Physical Nails: Summary

Now we will look at DFT guidelines specific to the design of boards containing non-boundary scan clusters and the special case of Random Access Memory (RAM) and Programmable Logic Device (PLD) clusters (in-system configuration).

Testing non-Boundary Scan Clusters

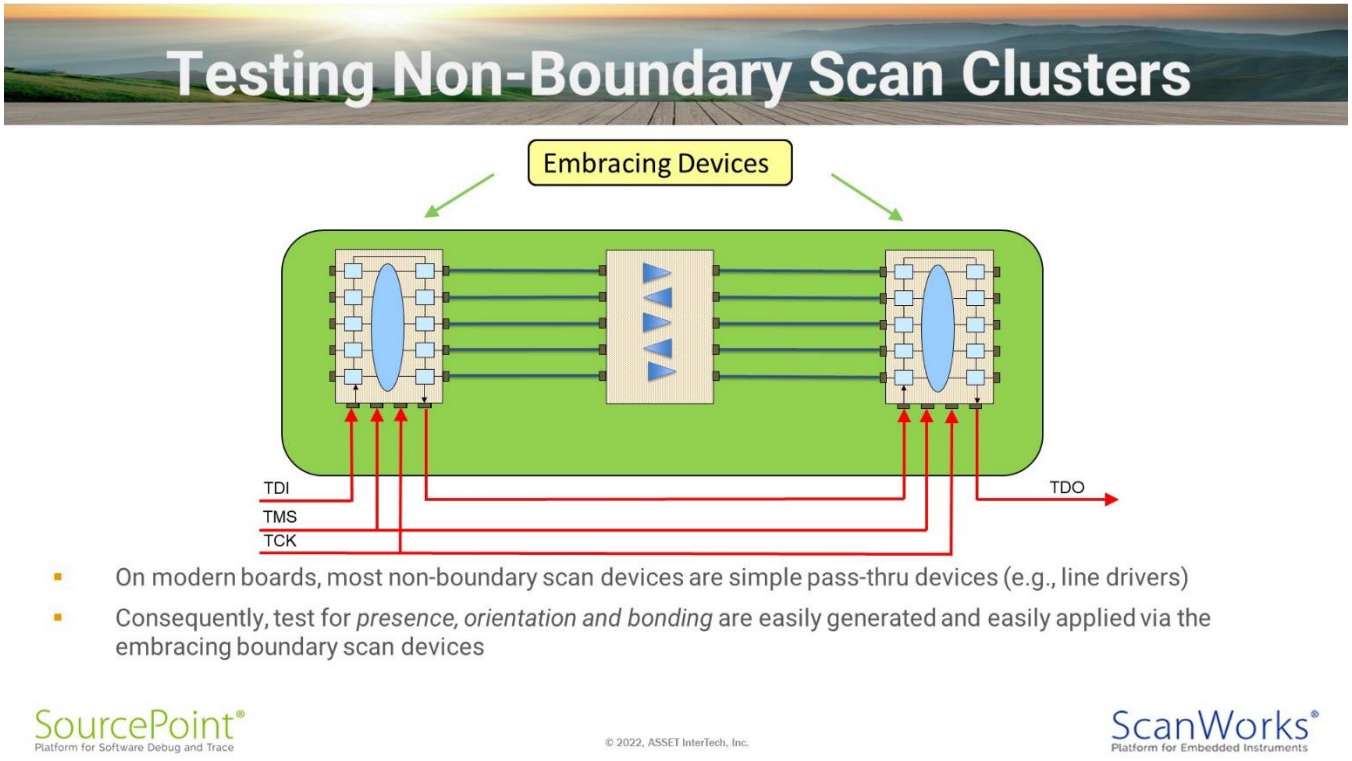


Figure 29: Testing Non-Boundary Scan Clusters

On most modern boards, the only non-boundary scan devices are simple line drivers (buffers), with or without inversion, or rerouting devices such as multiplexers. These devices are known as "pass thru" devices. It is a simple matter to generate presence, orientation and bonding tests for such devices and then apply the tests via the embracing boundary scan devices.

But, on older boards, there may be non-boundary scan MSI devices (i.e., devices with more complex functions, such as flip-flops, counters, shift registers, etc.) The next slide discusses how to handle such devices.

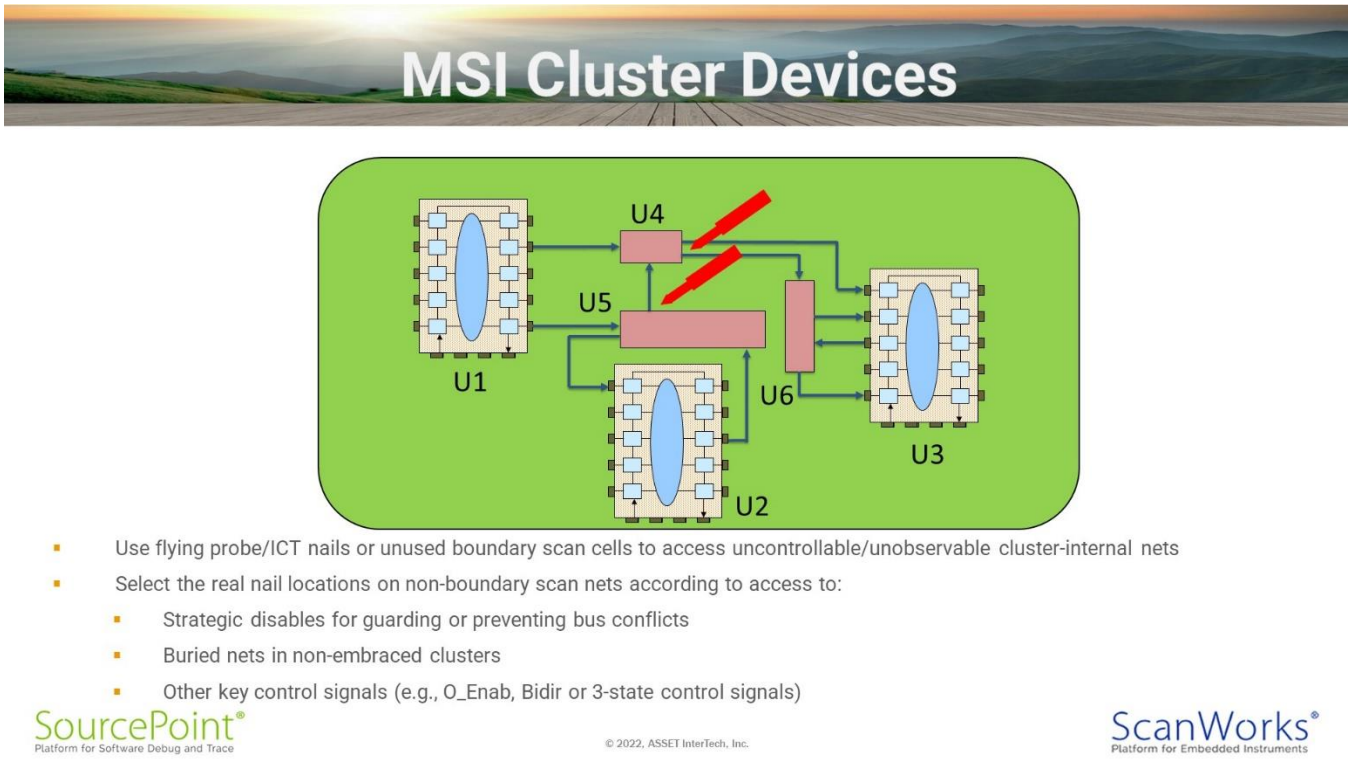


Figure 30: MSI Cluster Devices

Testing non-BS MSI devices for opens and shorts via a boundary scan interface (cluster testing) may not achieve 100% stuck-at and 2-net short fault coverage. Patterns for the non-boundary scan clusters can be taken from the extensive libraries of ICTs and validated via a fault simulator.

To maximize fault coverage on non-BS MSI devices, ensure maximum access to their pins either via boundary scan registers or direct from the primary connection to the board, or by using real nails (from a flying probe or bed-of-nails fixture).

If the board is to be tested using a mix of real nails (from a flying probe or bed-of-nail fixture) and virtual nails (from boundary scan cells), choose the selection of the real nail access nets carefully (i.e., where they will contribute the most to additional fault coverage). Vendors have access analysis tools to assist in the selection process. The selection process will also impact physical layout, causing certain nets to be brought to the surface of the board for physical probe purpose.

Where possible, provide direct access to key control signals on non-BS devices so that they can be easily configured into the correct state during test. If direct control is not possible, provide indirect control from an unused boundary scan cell.

Non-Participating BS Devices During Cluster Test

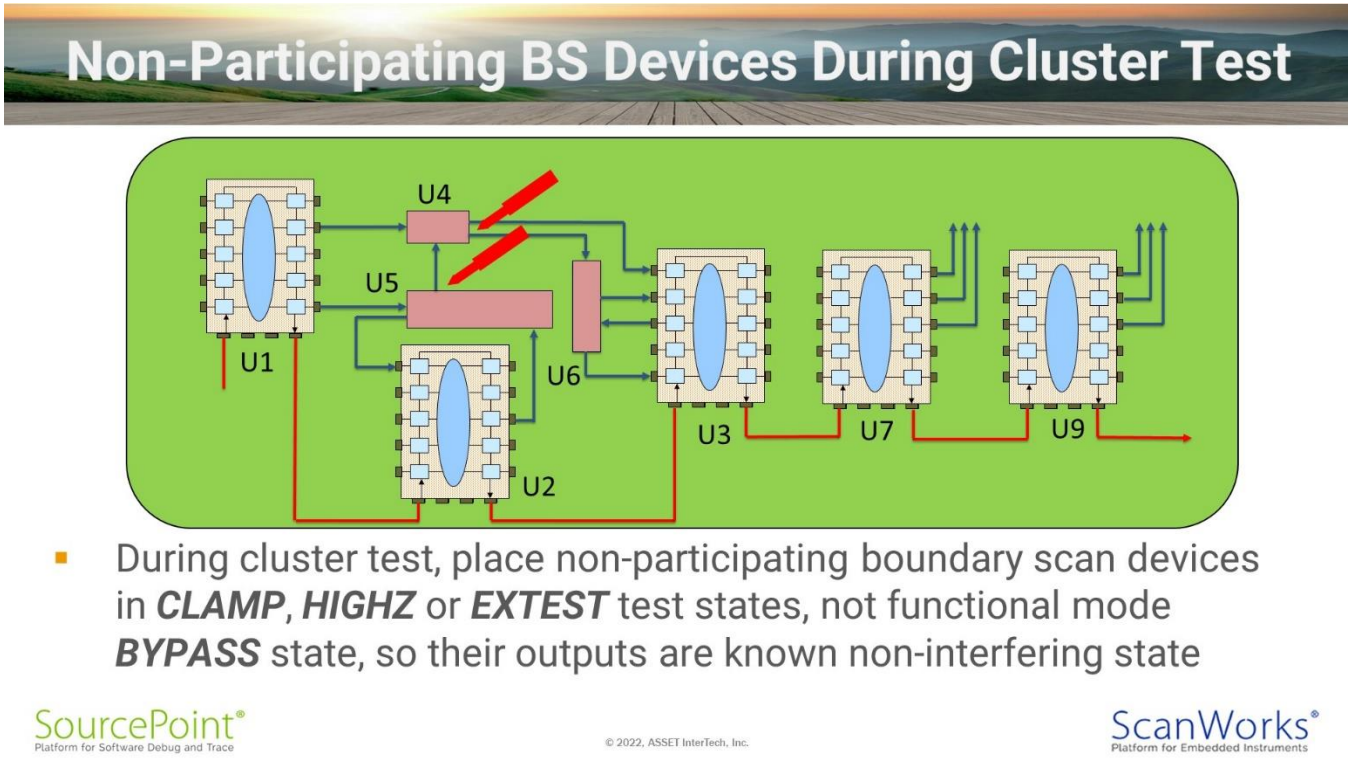


Figure 31: Non-Participating Boundary Scan Devices During Cluster Test

Boundary scan devices that do not participate in cluster testing should be placed into known safe states. One way to do this is to hold them in a test state rather than a functional state. The CLAMP, HIGHZ and EXTEST instructions can all be used to achieve this objective.

Special Case of Testing Memory Devices

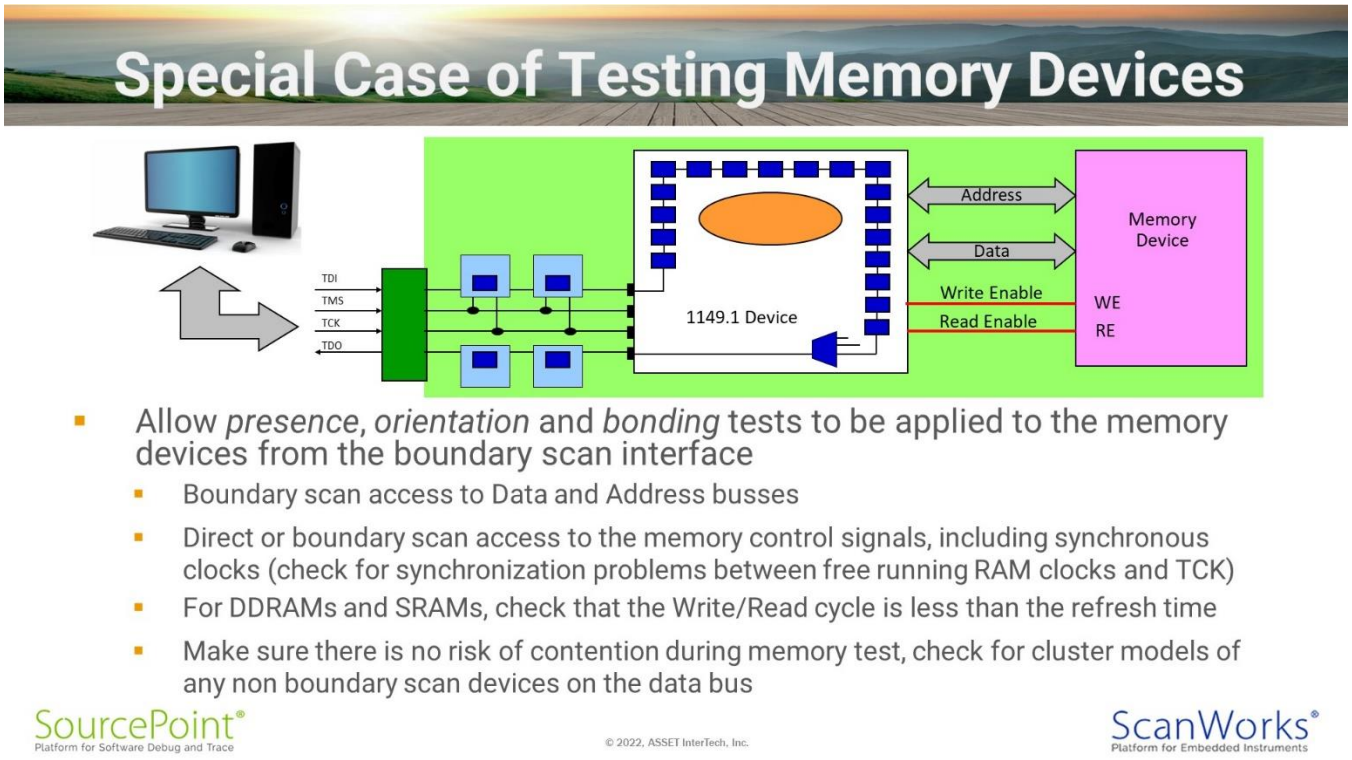


Figure 32: Special Case of Testing Memory Devices

A boundary scan interface can be used to test the presence, orientation, and bonding of onboard RAM devices. The tests require boundary scan access to the Data, Address and Control lines of the RAM. Commercial PC-based test systems support this use of boundary scan registers.

In the case of RAM devices that are not accessible from boundary scan devices, 1149.1-compliant buffer devices can be used to restore boundary scan access. National Semiconductor and Texas Instruments make 1149.1-compliant buffer devices for use on board internal busses such as the TI Octal and WidebusTM devices.

Use these devices for buffering bus signals rather than non-BS buffer devices. In the case of the TI devices, the boundary scan registers can be set up to become a pseudo random pattern generator (output scan cells) and CRC data compactor (input scan cells). A typical example of such a device is the SN74LVTH18502A Widebus™ Universal Bus Transceiver.

Memory Test: External Control

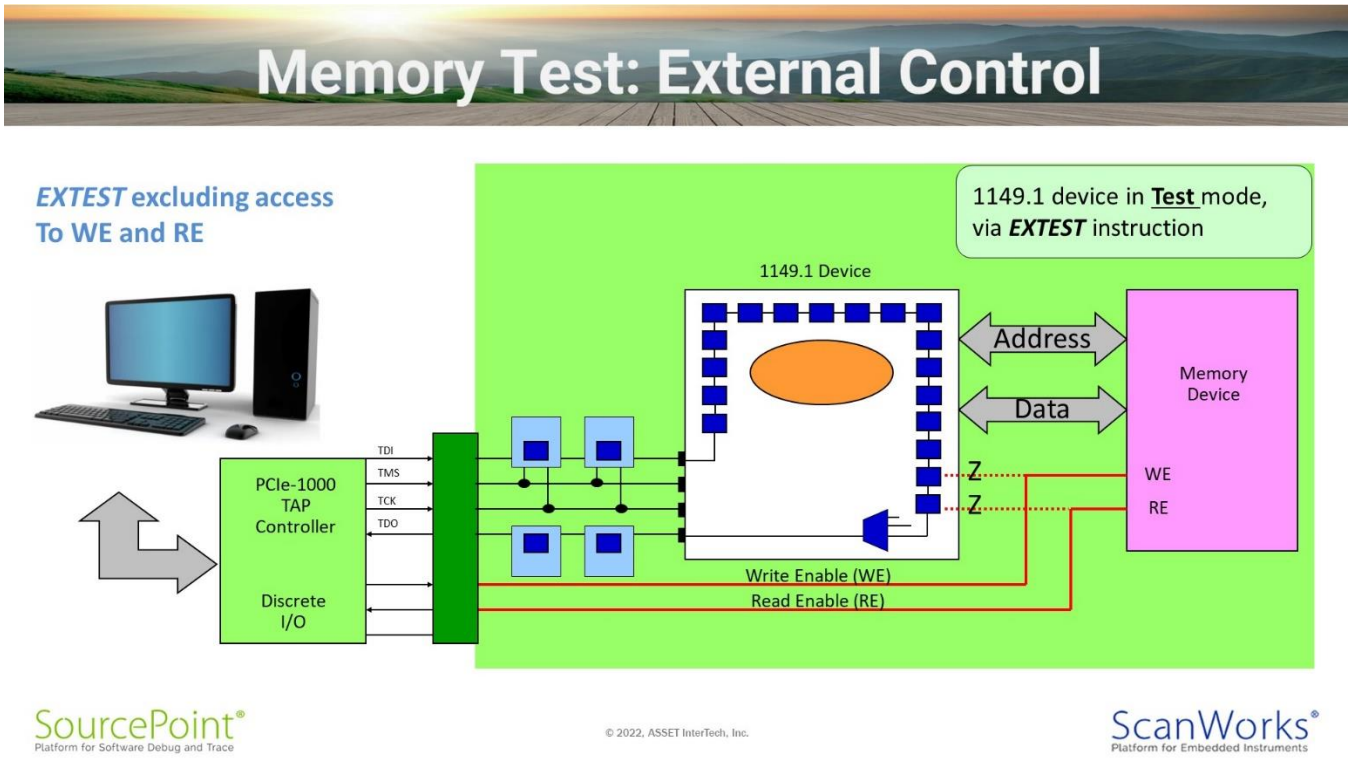


Figure 33: Memory Test: External Control

Here we see that the WE and RE control signals have been brought out to an edge connector position to allow programmable I/O pins (from the tester) to provide the control signals. This reduces the time it takes to check the presence, orientation, and bonding of the memory device.

If you do this, make sure that there is no damage caused by back driving to the output drivers of the normal source of the control signals. If there is the potential for damage, design the WE and RE sources to be tristate sources and place in high-Z state during the test mode, as shown.

In-System Configuration of CPLDs

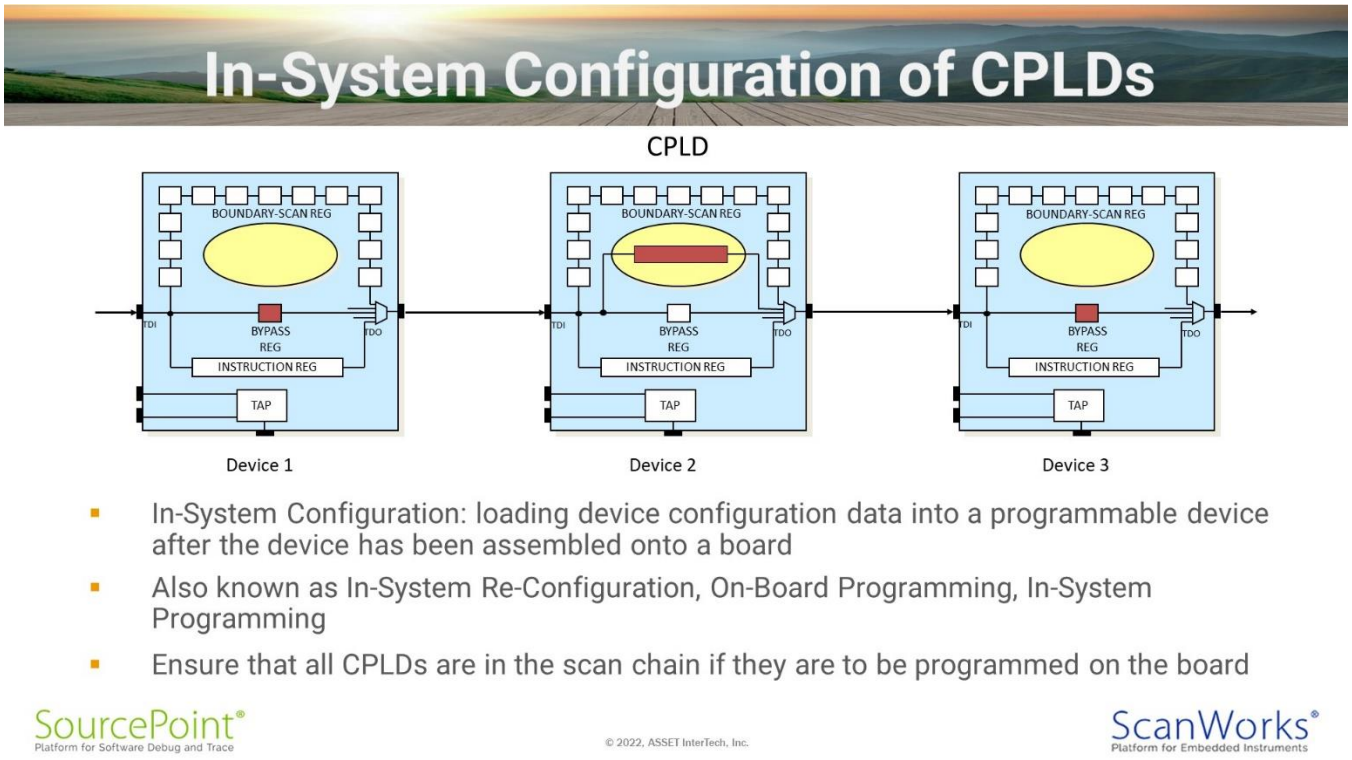


Figure 34: In-System Configuration of CPLDs

In-System Configuration (ISC), or In-System Programming (ISP) as it is often known is an application of 1149.1 boundary scan. ISC is the ability to load configuration data into a Complex Programmable Logic Device (CPLD), Field Programmable Gate Array (FPGA) or even a Flash device while the devices are mounted on a board.

The benefits of ISC are:

- simplifies inventory management
- reduces or removes the need for offline programming stations
- enables rapid prototype configuration and reconfiguration, thereby increasing design flexibility
- removes the need for onboard sockets which are often a cause of pin damage
- reduces risk of damage caused by mechanical handling and electrostatic discharge leading to improved quality of parts

- allows just-in-time programming (also known as design for postponement) and late changes (e.g., choice of language, personal details (SIMM cards), etc.)
- and allows program upgrades for system and field service debug

The programming of the CPLD device is conducted via the board level boundary scan access path, that is, from the edge connector through surrounding devices to the programmable device. Surrounding boundary scan (and non-boundary scan) devices must be placed in a safe state so as not to interfere with the in-system programming process. Boundary scan devices are first preloaded with safe values (using the PRELOAD instruction) and then placed in Bypass register mode using either the HIGHZ or CLAMP instruction. Placing surrounding boundary scan devices in bypass register mode also facilitates rapid access to the programmable device.

Boundary Scan Access to Programmable Devices

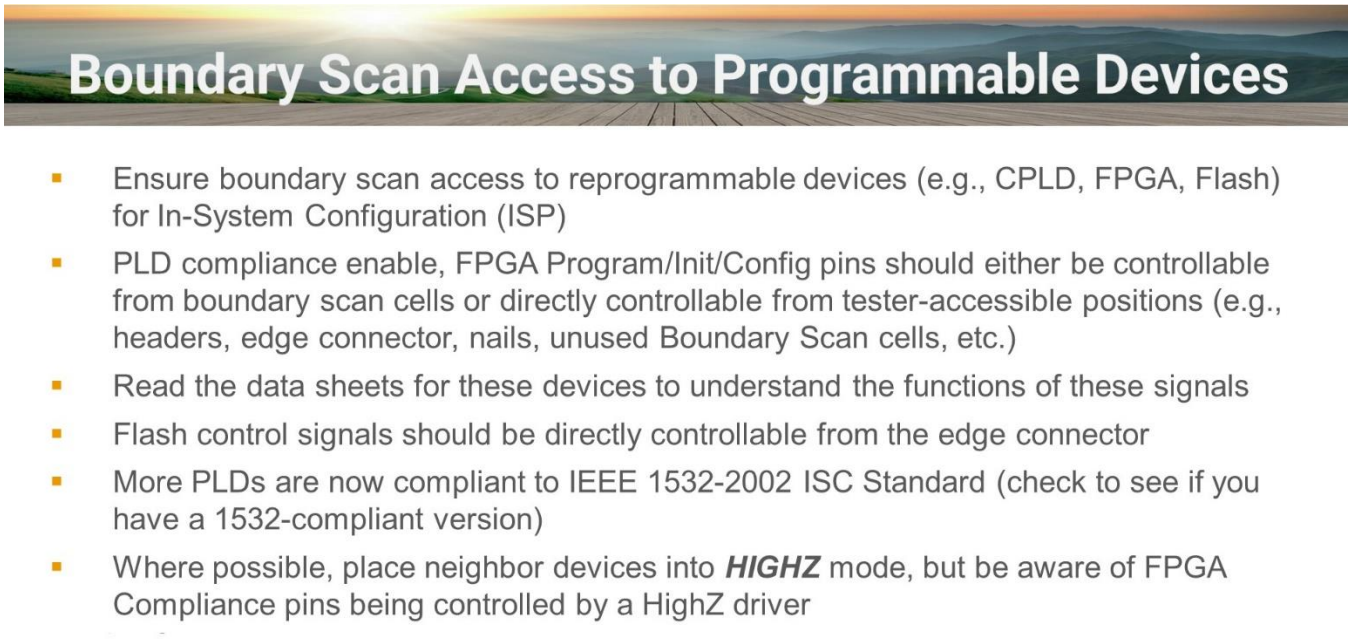


Figure 35: Boundary Scan Access to Programmable Devices

If the board contains programmable devices, such as CPLDs or FPGAs, make sure that the devices can be programmed, and reprogrammed, from a boundary scan interface. Ideally, such devices should be compliant to the IEEE 1532-2002 In-System Configuration Standard.

All significant control signals that control the operational status of onboard devices must be directly controllable when the board is in test mode. This includes board Power-on Self-Test, Boot or Program signals (e.g., Power-down, lnit, Reset, PRGM_, BOOT_).

Compliance Enable Pin Control

Figure 36: Compliance Enable Pin Control

Xilinx xc2s200: an example of a programmable device that has compliance pins to establish the 1149.1 logic. The two pins, PROGRAM and PWDNB, must both be held at logic 1 to establish the boundary scan logic.

Controlling PLD Compliance Pins

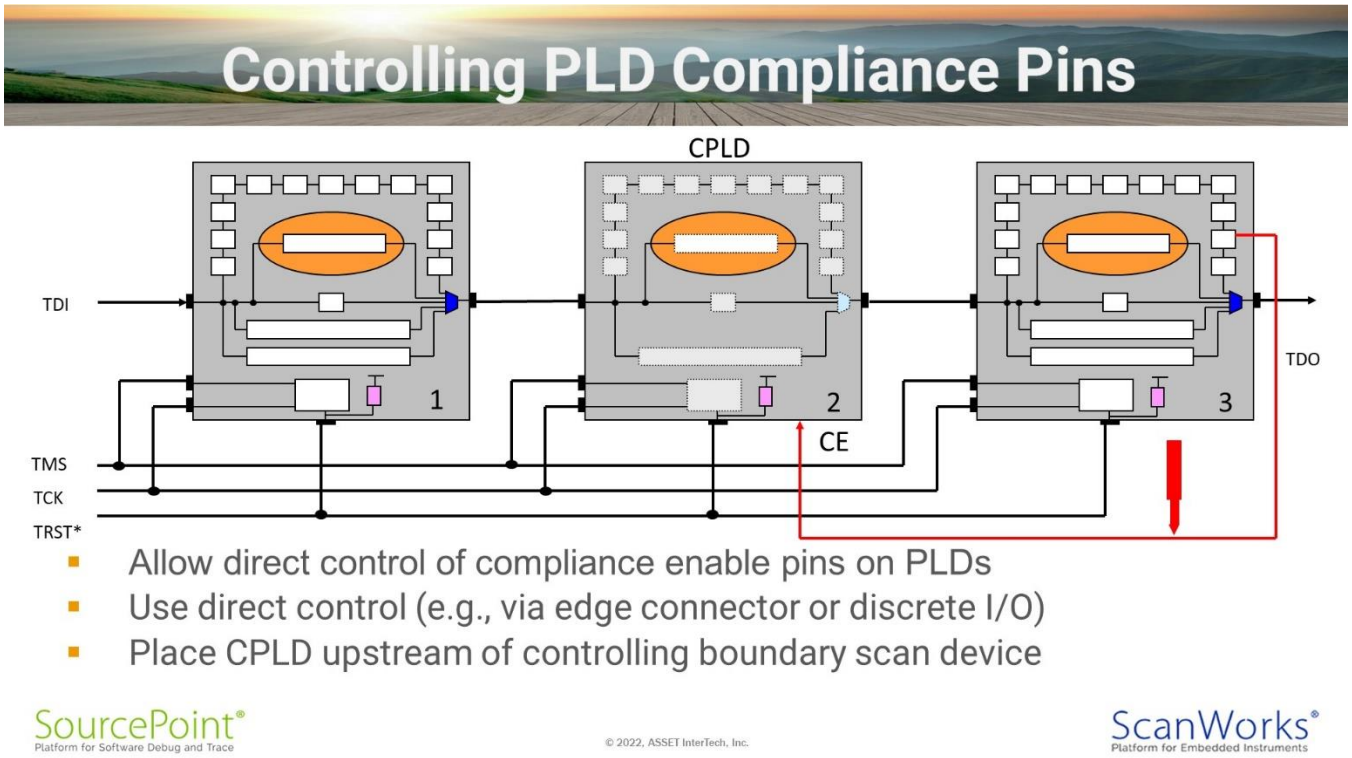


Figure 37: Controlling PLD Compliance Pins

Do not place control of compliance enable pins downstream of the programmable device. If you do this, the chain cannot be established. One solution is an automatic POR circuit on the board that has control of the compliance enable pins. Another solution is to place the programmable device downstream of the controlling device. This will not work if there is a defect that prevents the upstream devices from being correctly chained.

You should provide independent control of the compliance enable signal (e.g., through a physical nail or external connection: not through an unused boundary scan cell). Beware also the "blind apply" (e.g., if the scan chain order of the devices above are switched to 1-to-3-to-2, then it can be argued that the upstream path 1-to-3 can be set up and used to control the compliance-enable pins of the downstream device 2). This is true but assumes that there is no problem with the 1-to-3 path plus there is no problem with unknown values being presented to device 2's compliance enable pins during the initial set-up phase. It is much better to allow direct control on the compliance enable pins.

Flash Programming: 1149.1 EXTEST

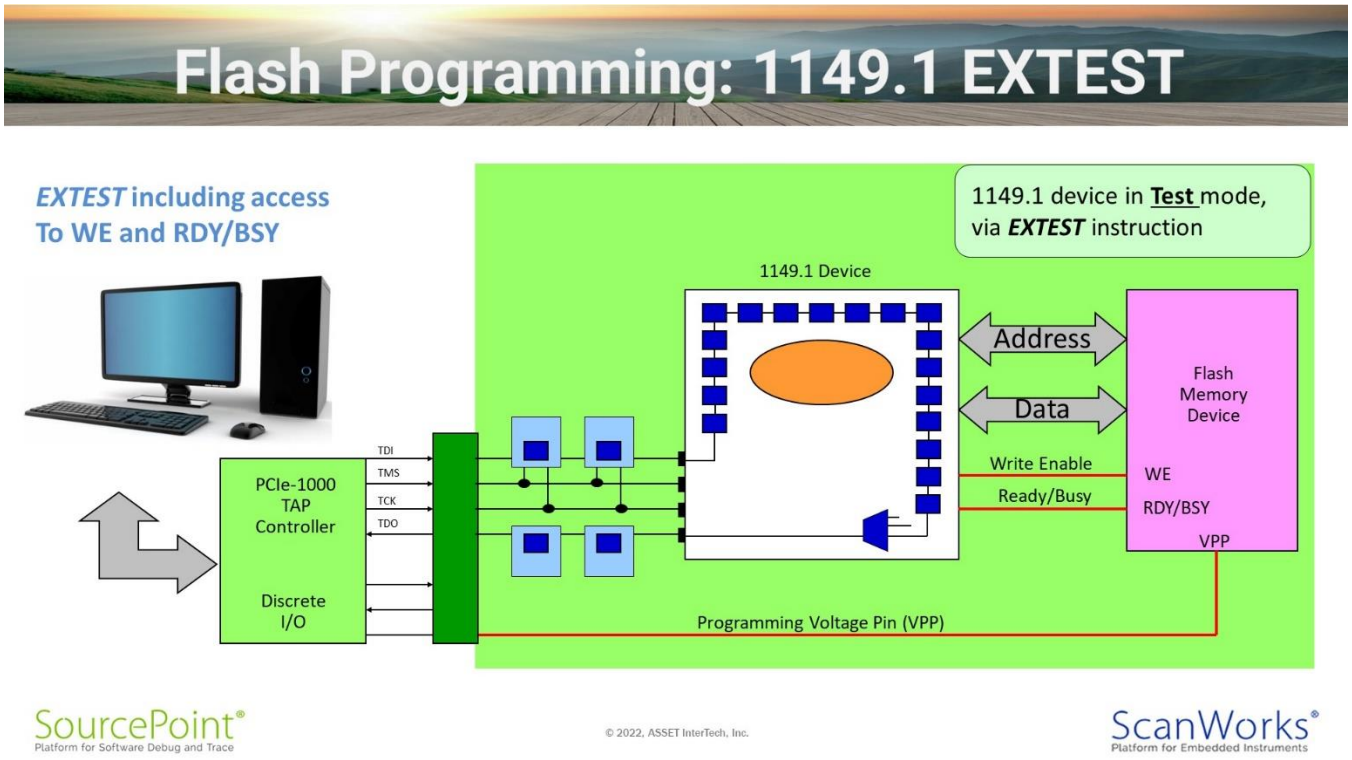


Figure 38: Flash Programming: 11491. EXTEST

The next few slides take a closer look at the ability to program flash memory devices through the boundary scan registers of adjacent devices connected to the flash Address and Data pins.

Programming flash devices in this way has become popular, especially amongst high volume consumer product vendors, such as cellular manufacturers. The slide above shows a basic system in which all access to the flash is from the boundary scan register of a single Application Specific Integrated Circuit (ASIC), but it would take too long to change the Write Enable values though the boundary scan register. The next slide shows a scheme where access to the flash Write/Read control pins is direct.

Over voltages, such as VPP, can either be supplied direct or can be provided by onboard FETs which are themselves controllable via a boundary scan cell, preferably in the same device that is in EXTEST mode to program the Flash device.

Flash Programming: External Control

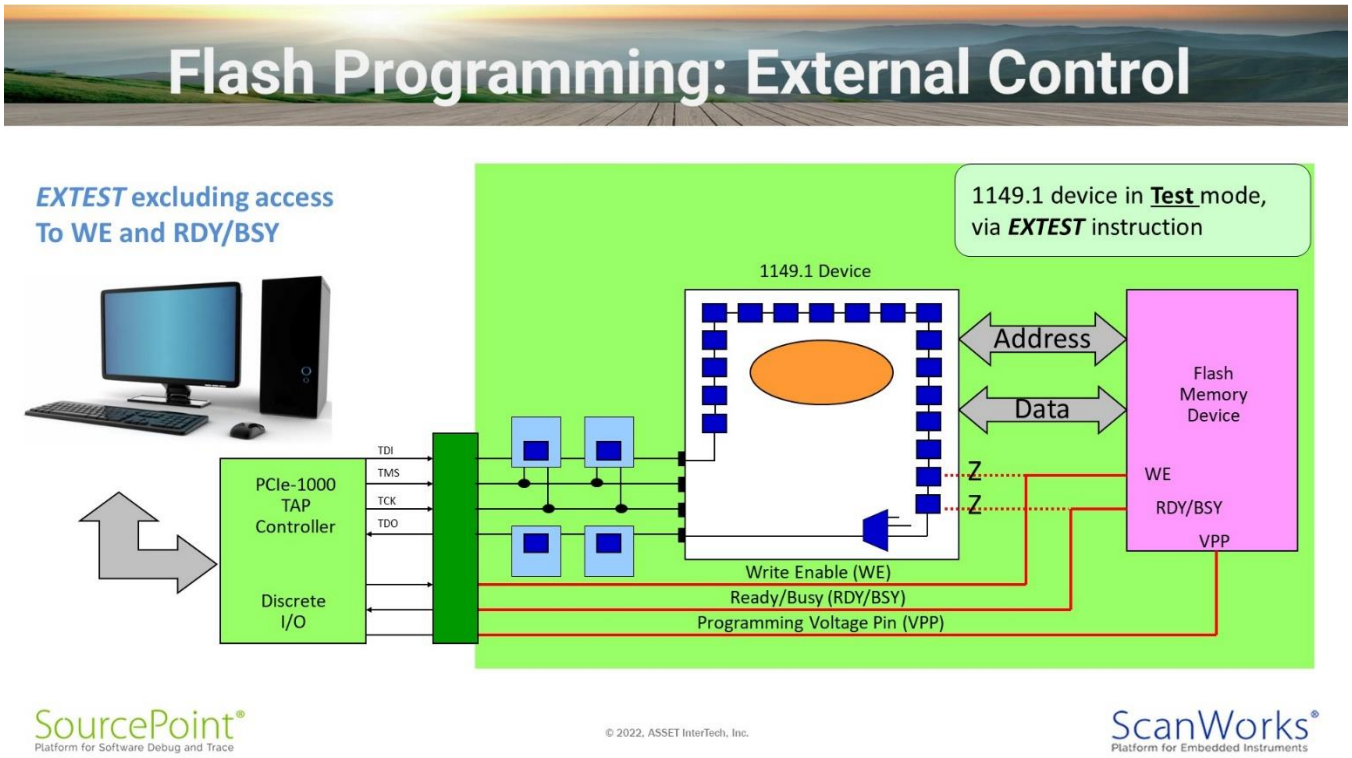


Figure 39: Flash Programming: External Control

Flash Programming Set-up and Constraints

Boundary scan programming device is in EXTEST mode. All other devices are in BYPASS or CLAMP/HIGHZ mode. All output pins must be controlled to safe values. Flash programming control signals, such as Write Enable, Ready/Busy and Over Voltage VPP pins, are controlled directly from the Discrete I/O pins of the interface pod.

For Write: address and data information is shifted into the boundary scan register (Shift+ Update).

For Read: data is shifted back into the boundary scan register and shifted out (Capture+ Shift).

Write/Read time: Function (TCK; length of the boundary scan register; indirect/direct access to WE, RDY/BSY; availability/non-availability of VPP).

WE signal: It takes three full scan loads to toggle WE high-low-high while holding the address and data steady. If we control WE outside boundary scan, we cut the number of scans by 3: one scan for the address and data, then toggle WE virtually instantaneously.

RDY/BSY signal: Flash programming is a charge pumping technology, the timing of which is not precise. Many Flash devices have a Ready/Busy pin to say when a byte/word has completed programming. The alternatives are: 1) interrogate the status byte of the device through boundary scan (yet more cycles) or 2) wait the maximum time for which programming is guaranteed to have completed.

Monitoring RDY/BSY directly allows the tester to program the next word/byte immediately the previous one has completed.

Breaking up a Single Scan Chain

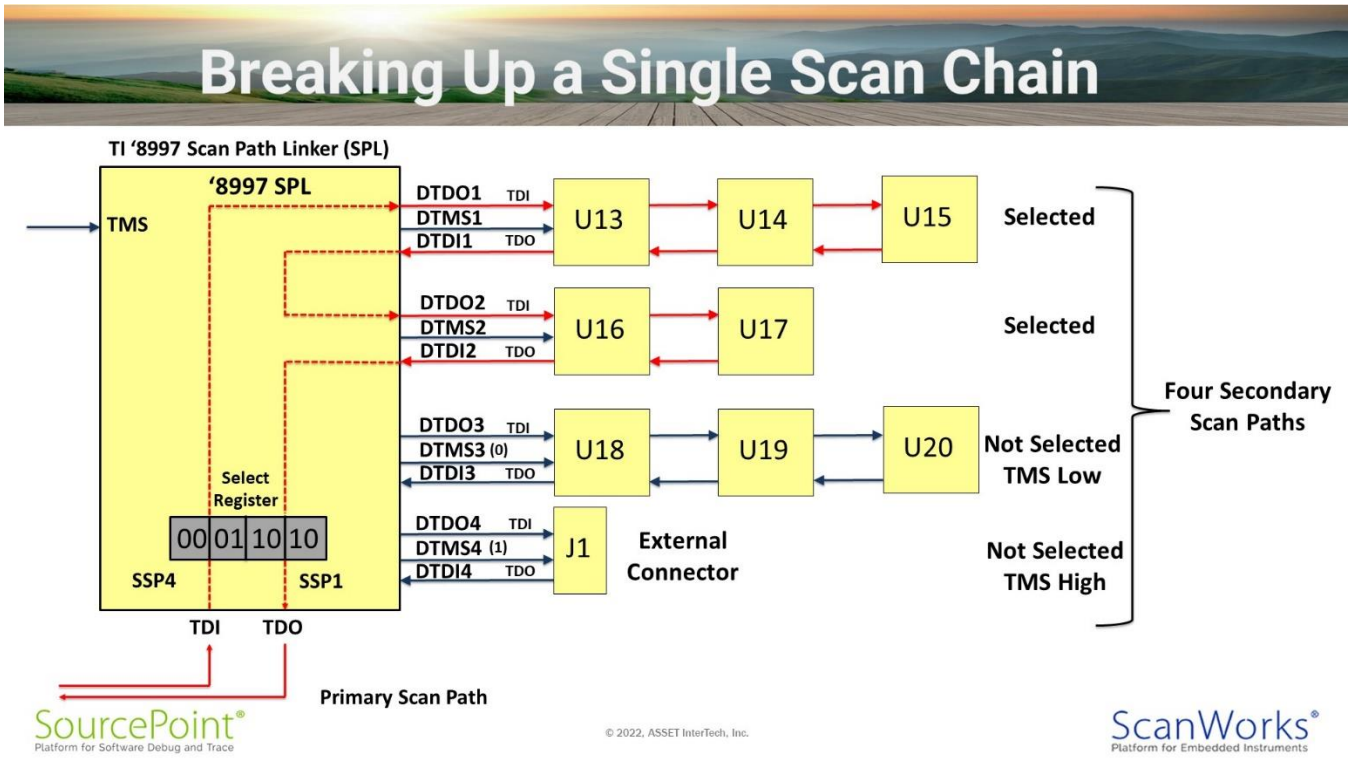


Figure 40: Breaking Up a Single Scan Chain

The slide shows a device from Texas Instruments called the '8997 Scan Path Linker (SPL). This device allows dynamic selection of secondary scan chains to link with the primary scan chain. Such a device

can be used to select, or otherwise, a subset of a main scan chain for a specific reason (e.g., direct access to a flash device for programming purposes).

The Secondary Scan Paths (SSPs) are either included or excluded from the Primary Scan Path based on the configuration loaded into an SPL internal register called the Select Register, selected by the SCANSEL instruction. In the example above, SSP1 and SSP2 are selected and linked together. SSP3 and SSP4 are excluded from the scan chain. The order of the scan chain is Primary TDI to SSP1 to SSP2 back to primary TDO.

Accessing a Flash/PLD Device

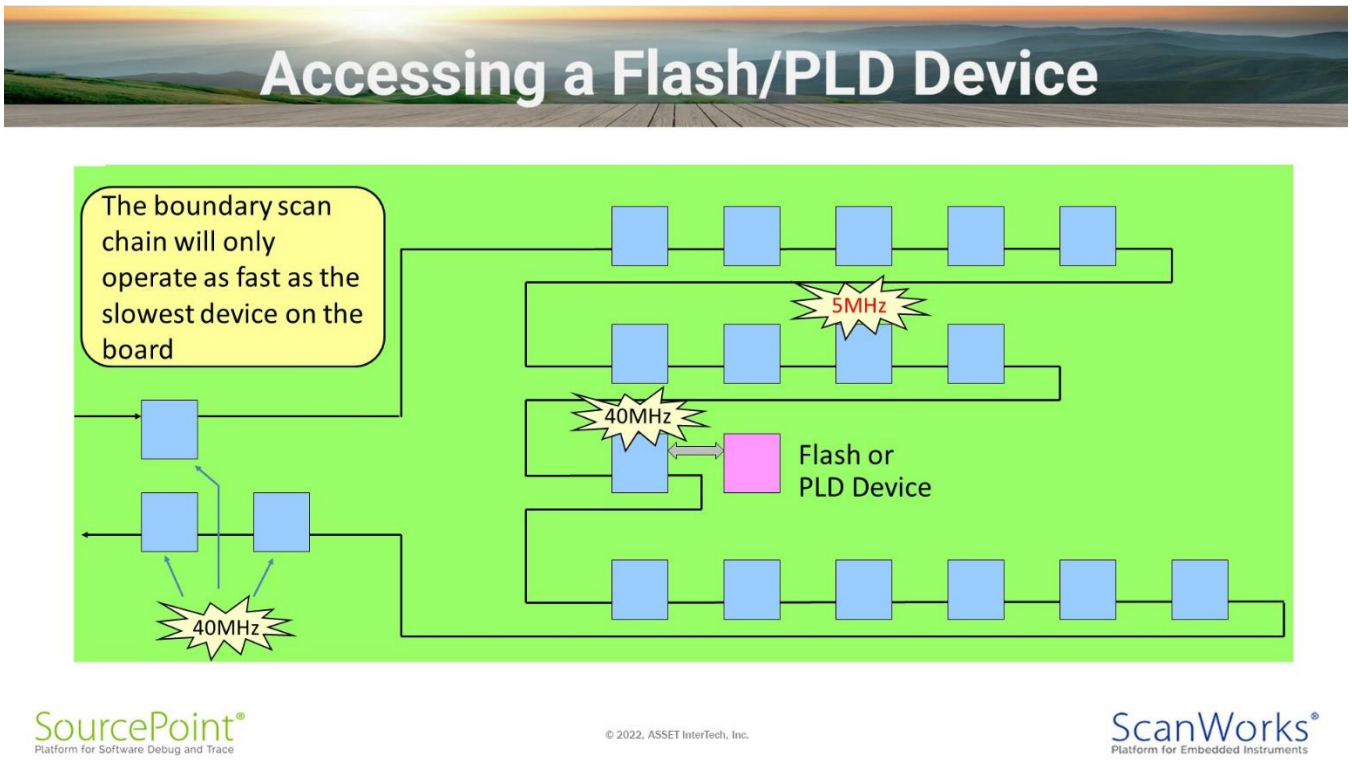


Figure 41: Accessing a Flash/PLD Device

Here we see a Flash (or PLD) device accessible from an ASIC/boundary scan device. Let us assume that we wish to program this device at the maximum rated frequency of 40MHz - the max TCK for the ASIC. One problem however is that TCK can only be as fast as the slowest device in the chain. If the ASIC can accept a TCK of, say, 40MHz but another device in the chain is only able to run at a TCK of,

say, 5MHz, then the maximum speed of the whole active chain is limited to 5MHz. This may produce an unacceptable reduction in the flash programming time.

Direct Access to a Flash/PLD Device

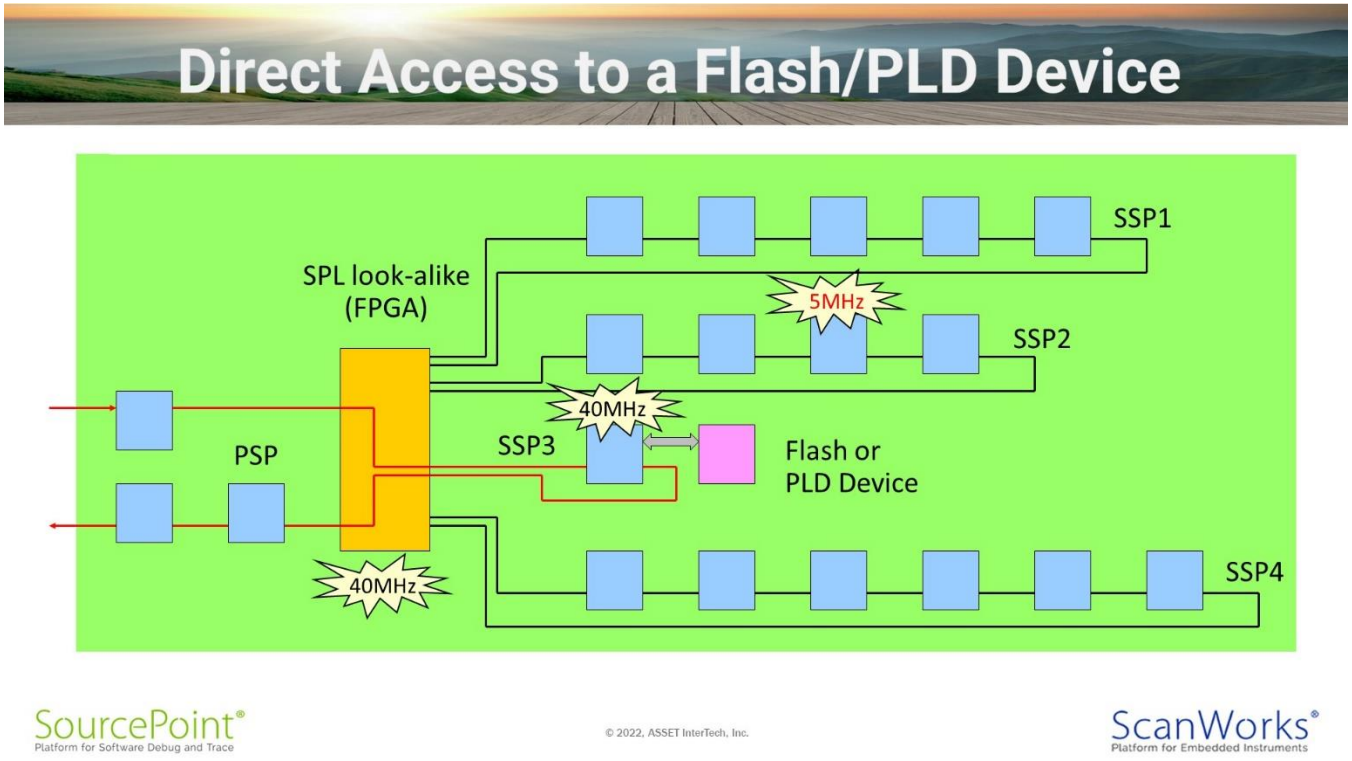


Figure 42: Direct Access to a Flash/PLD Device

A solution is to use a 40MHz TCK FPGA programmed to behave like a Texas Instruments ‘8997 Scan Path Linker (SPL) or National Semiconductors Enhanced ScanBridge device to exclude the slow boundary scan devices in the chain, as shown. The SPL has four secondary scan paths, each individually selected through a special SPL configuration instruction called SCANSEL. The ScanBridge device has three secondary scan ports, again each individually selectable.

A limitation with the TI and National Semiconductor devices is that they work at low maximum TCK frequencies: 20MHz for Tl's devices and 25MHz for National's devices. This was why the suggestion above is based on an FPGA look alike. Another concern is that the TI and National Semiconductor devices are currently 5V devices whereas the boundary scan devices in the chain may be working at lower supply voltages. The reader should check with the suppliers of these devices to see if lower

voltage devices are available. Alternatively, check the product offerings of Lattice (ispGDX family) or Firecron. An alternative solution is to use removable/replaceable jumpers to bypass all boundary scan devices in the chain except the Flash or PLD devices, see the emulation slide for an example of the use of jumpers.

Flash Programming Through an EFC

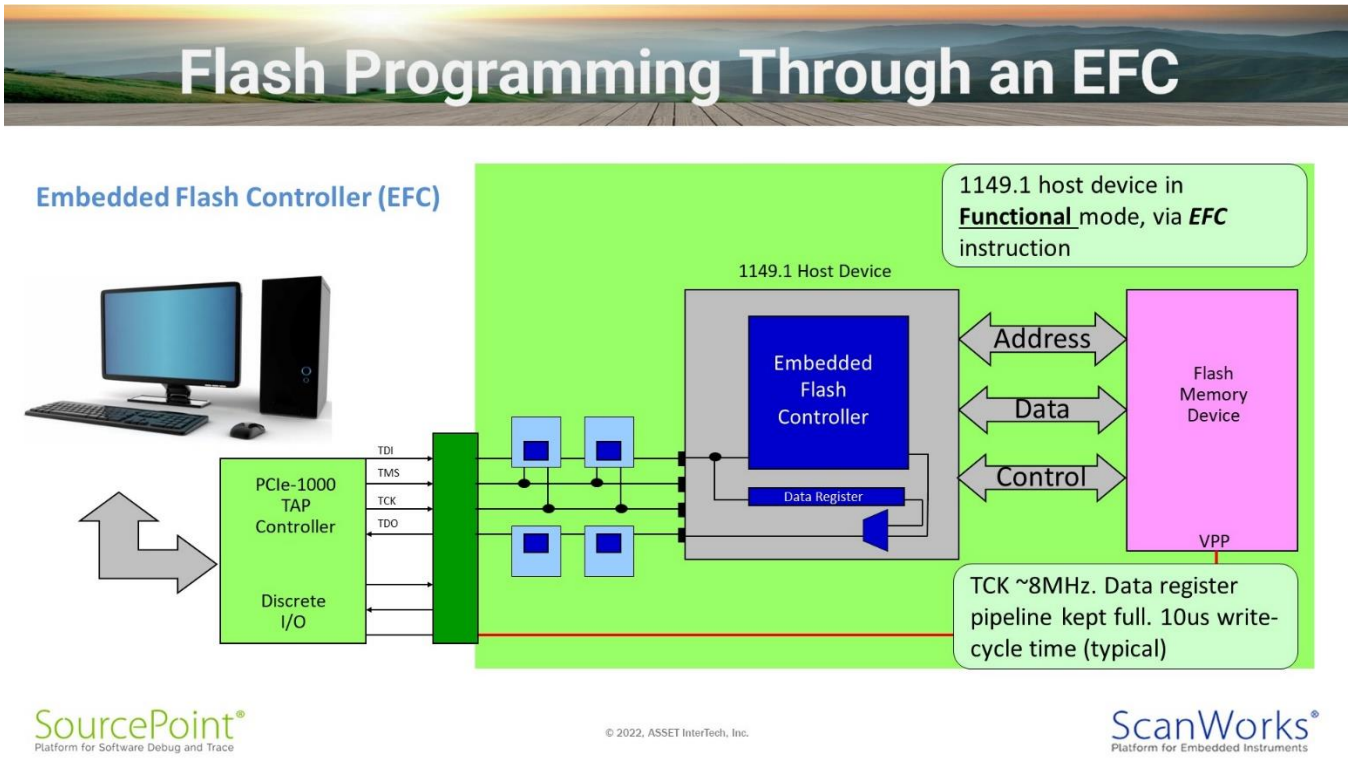


Figure 43: Flash Programming Through an EFC

Another solution, embed an Embedded Flash Controller (EFC) inside a host 1149.1 device, controlled via the device's 1149.1 structures. The host device TDI-TDO provides a path to an internal Data register to load the initial Address and Data. Once initialized, the embedded controller implements the programming procedure for the Flash, including all necessary control signals for Write and Read back. Incrementing the Address can be automatic.

Advantages are:

- Flash writing can be at SysClk speeds, not TCK speeds.
- Flash data bus width can be at full system bus width

- One controller can service multiple flash devices
- The host 1149.1 device is in functional (safe) mode, not test mode
- Flash can be reprogrammed

Providing Isolation for Emulation

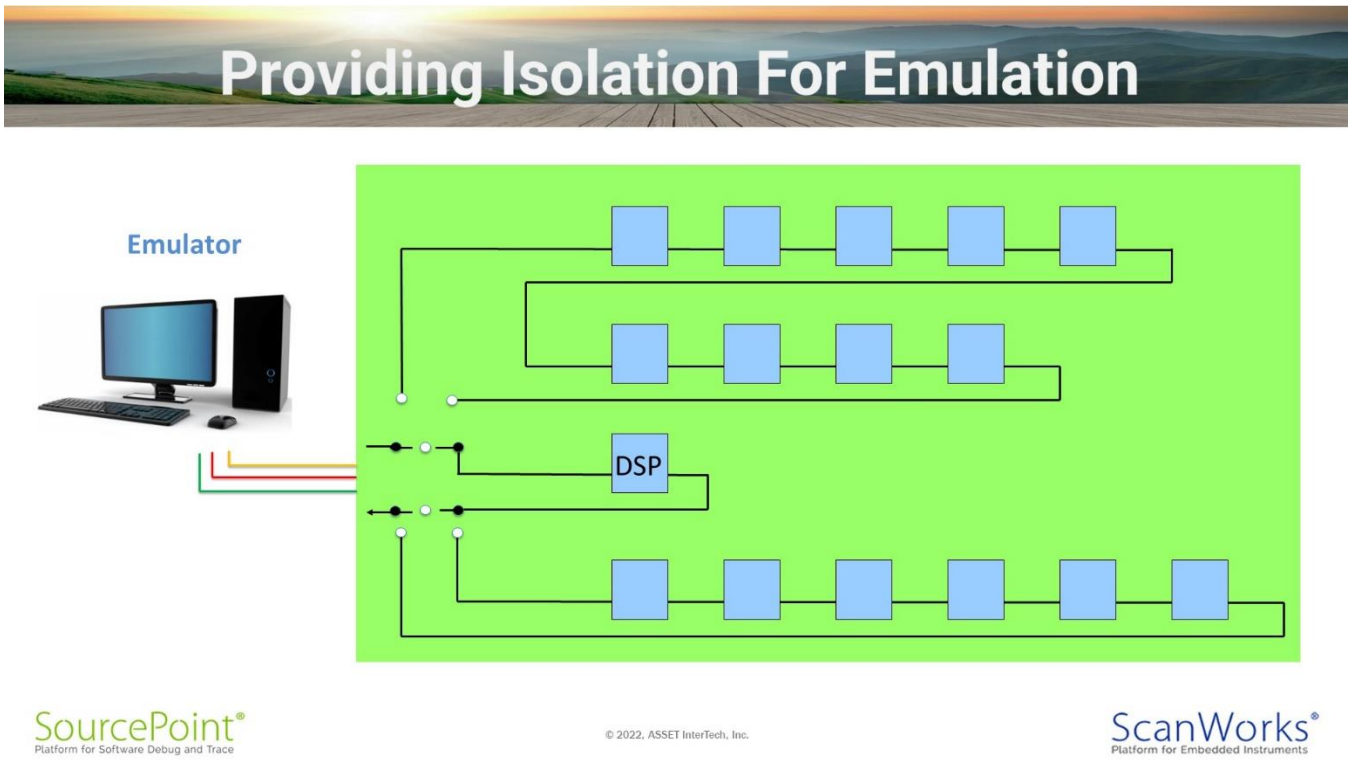


Figure 44: Providing Isolation for Emulation

Most device emulators (DSP, RISC Emulators) have a problem unless their device is the only device in the scan chain. If these types of devices are placed in a full board level boundary scan chain, this can cause emulation time and device isolation issues during development/software debug. It is better to provide an ability to isolate the emulation device entirely from the rest of the chain and so leave the other devices in functional mode to support the emulation process.

Ways to do this vary from simple jumper selections for TDI-TDO (as shown above) to a connector, all the way up to a multiplexer design selectable from a pin on a header that the emulator/tester plugs into, or a bridge device that supports pass through modes.

Conclusion

In this eBook, we examined Design for Test (DFT) guidelines specific to the design of boards to be tested through the boundary scan registers of IEEE 1149.1-compliant devices. Boards designed implementing boundary scan DFT guidelines are easy to test during the manufacturing process with a boundary scan test tool. The first guideline listed in this document is the most important; where possible, use 1149.1-compliant devices on the board. Next, to ensure testability through the boundary scan registers, the 1149.1-compliant devices must be connected TDI-to-TDO. The other device TAP signals, TMS, TCK, and TRST* (if applicable) are also connected in a distributed fashion. This configuration of the 1149.1-compliant devices is known as a scan chain. Boundary scan DFT guidelines state that access to the TAP signals be provided to the boundary scan test tool.

Boundary scan DFT increases the test coverage of the board by making more boundary scan registers accessible. The more boundary scan registers that are accessible on the board, the higher the boundary scan test coverage for detection of open and/or short structural faults. As the boundary scan technology matured, it is now used for testing of memory devices, configuration of programmable devices, and programing of flash devices. DFT guidelines apply to these use cases as well to ensure boundary scan access to these device types. There are also DFT guidelines that decrease the configuration and programming time of these devices via boundary scan. Device configuration and programming are time intensive operations during manufacturing, therefore, reducing time in this step is important.

Boundary scan DFT is a vital component of an overall board test strategy. If boundary scan DFT guidelines outlined cannot be implemented on the initial board design, they can be implemented on subsequent board revisions to incrementally increase coverage. Also, more 1149.1 devices can be substituted on the board in place of non-1149.1 devices, thus providing more boundary scan access which increases boundary scan test coverage. Maximizing board test coverage is the overall goal.

Maximizing board test coverage is imperative to improving manufacturing yields, increasing product quality, and reducing product returns. The prime reasons for developing a board test strategy are to find defects, diagnose the cause of the faults, and repair them quickly. The obtained fault data can be used to constantly improve the processes that produced the board and the faults in the first place. Following boundary scan DFT is an important aspect of the overall test strategy for companies who have committed to a non-intrusive board test strategy.