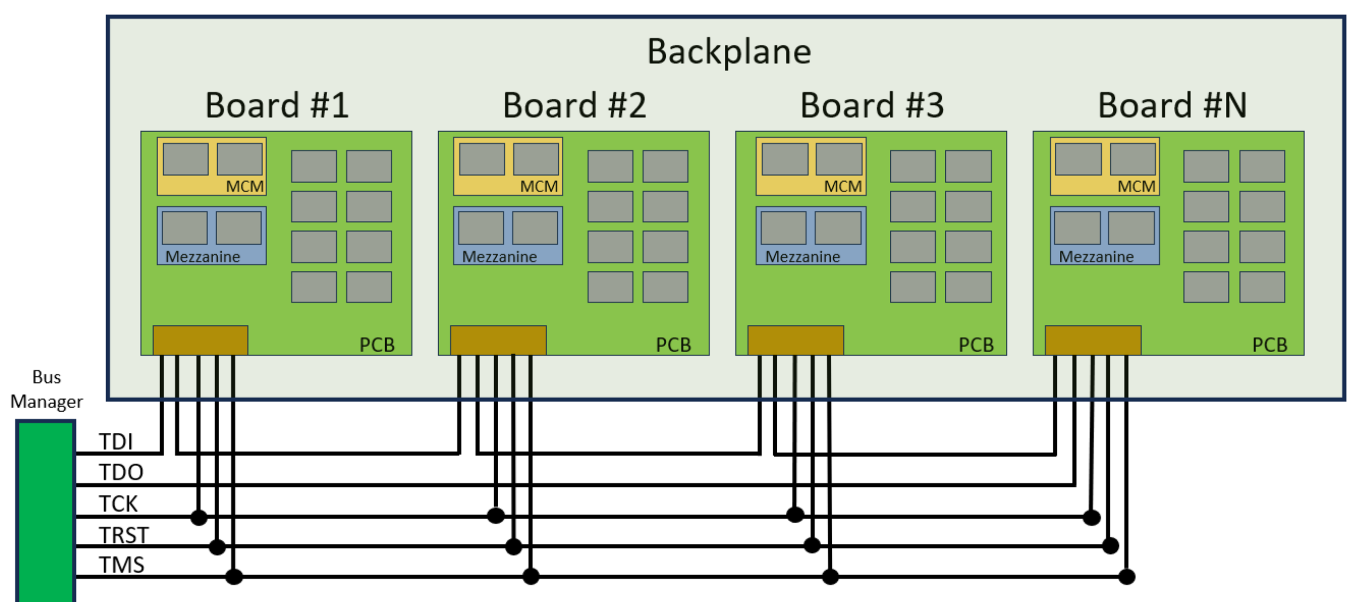
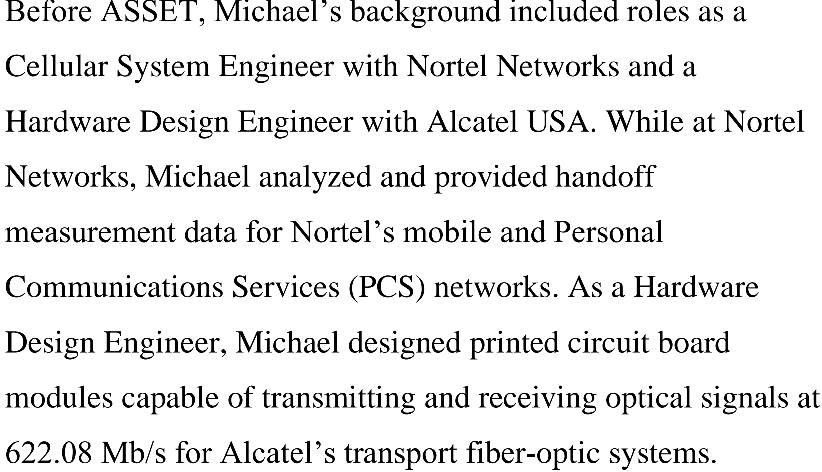
# Guidelines for System-level JTAG Design



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 and Professional Services organization. As Product Manager, Michael provides strategic direction for the ScanWorks BST product, ensuring an exceptional customer experience and adherence to current and future IEEE standards requirements. Michael coordinates with ASSET's cross-functional teams such as Marketing, Sales, Support, and Research and Development, to bring ASSET's business goals to fruition.



Before ASSET, Michael's background included roles as a Cellular System Engineer with Nortel Networks and a Hardware Design Engineer with Alcatel USA. While at Nortel Networks, Michael analyzed and provided handoff measurement data for Nortel's mobile and Personal Communications Services (PCS) networks. As a Hardware Design Engineer, Michael designed printed circuit board modules capable of transmitting and receiving optical signals at 622.08 Mb/s for Alcatel's transport fiber-optic systems.

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

# Table of Contents

Executive Summary ........ 1
Board DFT based on Boundary Scan – Volume 1 ........ 2
Board DFT based on Boundary Scan – Volume 2 ........ 3
What is System-level JTAG Design? ........ 4
What is System-level JTAG Design - SJTAG ........ 5
System-level JTAG Design Architectures - Ring ........ 6
System-level JTAG Design Architectures - Star ........ 7
System-level JTAG Design Architectures – Multi-drop Device ........ 8
System-level JTAG Design Architectures – SCANSTA112 ........ 9
System-level JTAG Design Architectures – Multi-drop Device Summary ........ 10
System-level JTAG Design Architectures – Multi-TAP Controller ........ 11
System-level JTAG Design Architectures – PCIe-410 ........ 12
System-level JTAG Design Architectures – Summary ........ 13
System-level JTAG Design – Embedded Applications – SED for Test ........ 14
System-level JTAG Design – Embedded Applications – SED for Test Advantages ........ 15
System-level JTAG Design – Embedded Applications – SED for Test Resources ........ 16
System-level JTAG Design – Embedded Applications – SED for Debug ........ 17
System-level JTAG Design Embedded Applications – SED for Built-in Self-Test ........ 18
Guidelines for System-level JTAG Design – Summary ........ 19
System-level JTAG Desing – Additional Resources ........ 20
Conclusion ........ 21

## Table of Figures


Figure 1: System-level JTAG Design – Agenda ..... 1
Figure 2: Board DFT based on Boundary Scan – Volume 1 ..... 2
Figure 3: Board DFT based on Boundary Scan – Volume 2 ..... 3
Figure 4: What is System-level JTAG Design? ..... 4
Figure 5: What is System-level JTAG Design – SJTAG ..... 5
Figure 6: System-level JTAG Design Architectures – Ring ..... 6
Figure 7: System-level JTAG Design Architectures – Star ..... 7
Figure 8: System-level JTAG Design Architectures – Multi-drop Device ..... 8
Figure 9: System-level JTAG Design Architectures – SCANSTA112 ..... 9
Figure 10: System-level JTAG Design Architectures – Multi-drop Device Summary ..... 10
Figure 11: System-level JTAG Design Architectures – Multi-TAP Controller ..... 11
Figure 12: System-level JTAG Design Architectures – PCIe-410 ..... 12
Figure 13: System-level JTAG Design Architectures – Summary ..... 13
Figure 14: System-level JTAG Design – Embedded Applications – SED for Test ..... 14
Figure 15: System-level JTAG Design – Embedded Applications – SED for Test Advantages ..... 15
Figure 16: System-level JTAG Design – Embedded Applications – SED for Test Resources ..... 16
Figure 17: System-level JTAG Design – Embedded Applications – SED for Debug ..... 17
Figure 18: System-level JTAG Design – Embedded Applications – SED for Built-in Self-Test ..... 18
Figure 19: Guidelines for System-level JTAG Design – Summary ..... 19
Figure 20: System-level JTAG Design – Additional Resources ..... 20

## Executive Summary

In this eBook, we will examine guidelines for system-level JTAG design. This eBook is the 3rd in a Design for Test (DFT) tome which has focused on the guidelines specific to the design of boards, and now systems, to be tested through IEEE 1149.1 JTAG/Boundary Scan. System-level JTAG testing ensures the boards within a system are structurally sound, it also ensures the complex system functions when assembled. Structural testing of boards, individually, is an important aspect of complete system functionality. But several types of faults can take place when the entire system is assembled. System-level JTAG is useful in testing backplane connections, connectors, and backplane passive and active circuitry. Testing systems, as a complex whole, can also be used during functional testing and environmental testing as well. There are several architecture designs which make system-level JTAG possible. Considerations of each design must be given forethought prior to implementation. Also, each architecture design has advantages and disadvantages of implementation. In addition to several architectures, there are devices, hardware, and embedded applications that lend themselves quite nicely to implementing a system-level JTAG design solution (Figure 1).

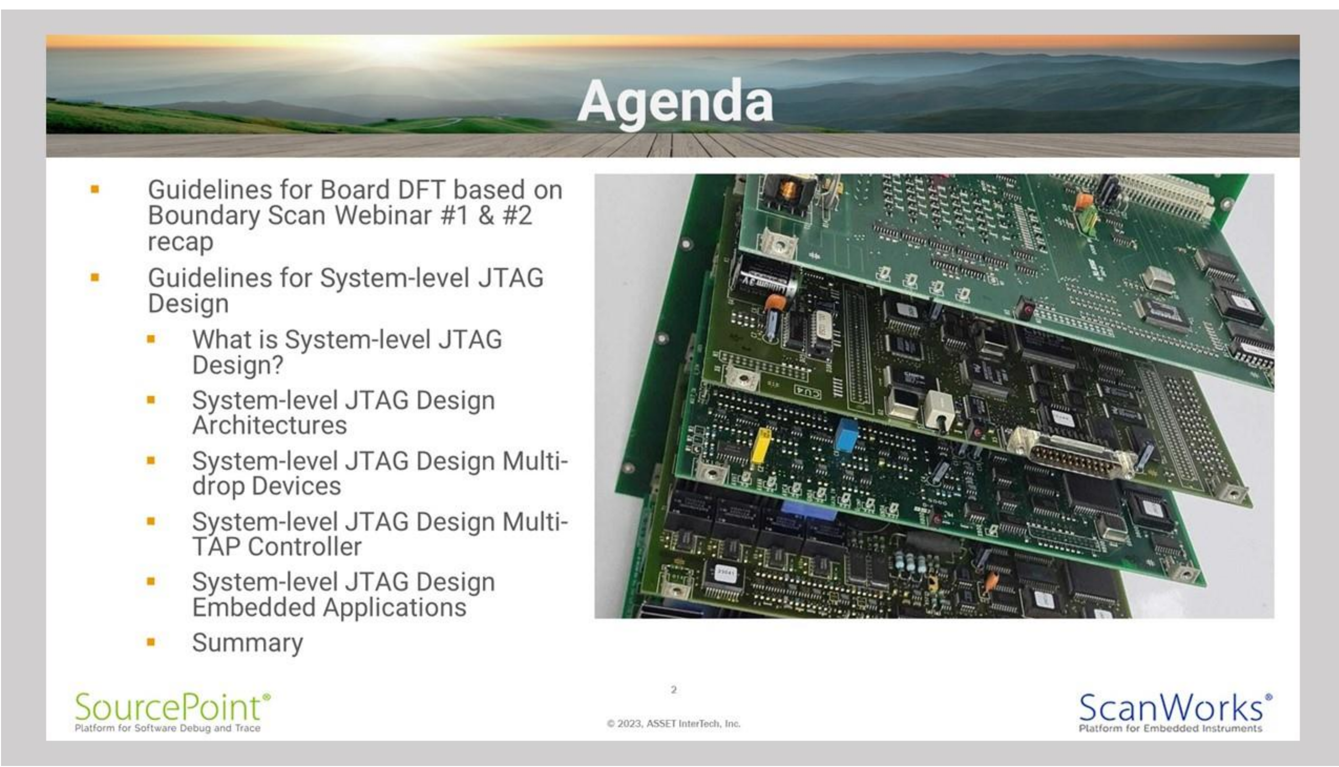



**Figure 1: System-level JTAG Design – Agenda**

## Board DFT based on Boundary Scan – Volume 1

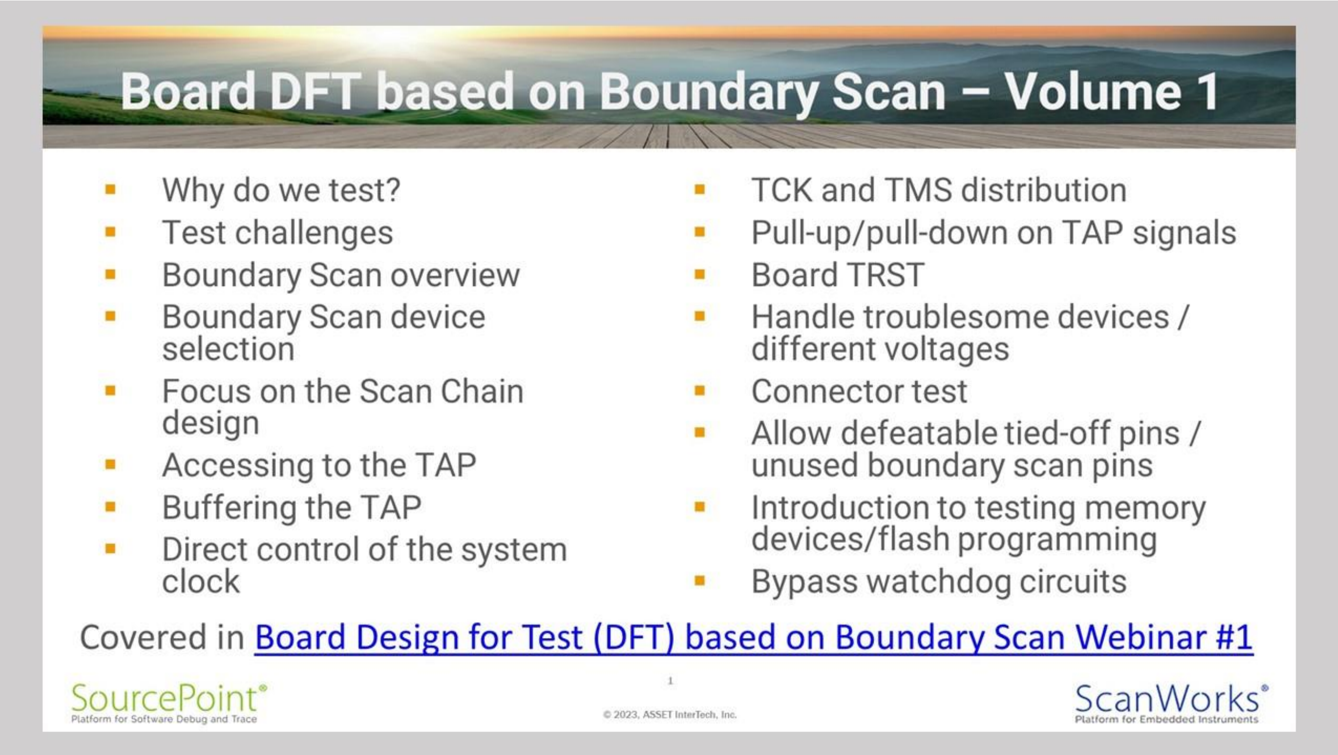



**Figure 2: Board DFT based on Boundary Scan – Volume 1**

These board-level DFT guidelines were covered in my 1st DFT eBook. This eBook can be accessed from the ASSET website (Figure 2).

## Board DFT based on Boundary Scan – Volume 2

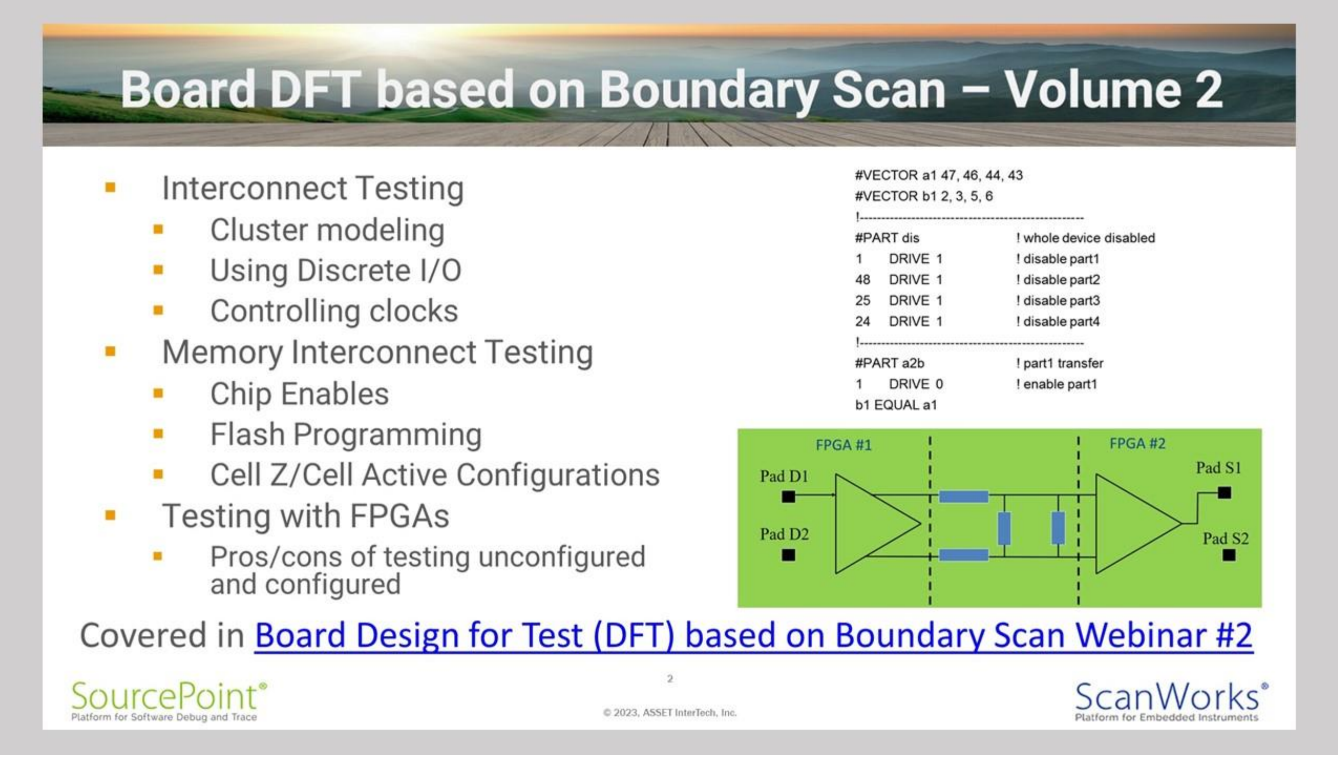



**Figure 3: Board DFT based on Boundary Scan – Volume 2**

These board-level DFT guidelines were covered in my 2nd DFT eBook. This eBook can also be downloaded from the ASSET website. The 1st two eBooks were focused on board-level for DFT for Boundary Scan, now we'll focus on implementing a system-level JTAG solution leveraging the on-board Boundary Scan resources along with architecture modifications to the backplane, the use of multi-drop devices, multi-Test Access Port (TAP) hardware, and embedded applications, to implement a complete system-level test solution (Figure 3).

## What is System-level JTAG Design?

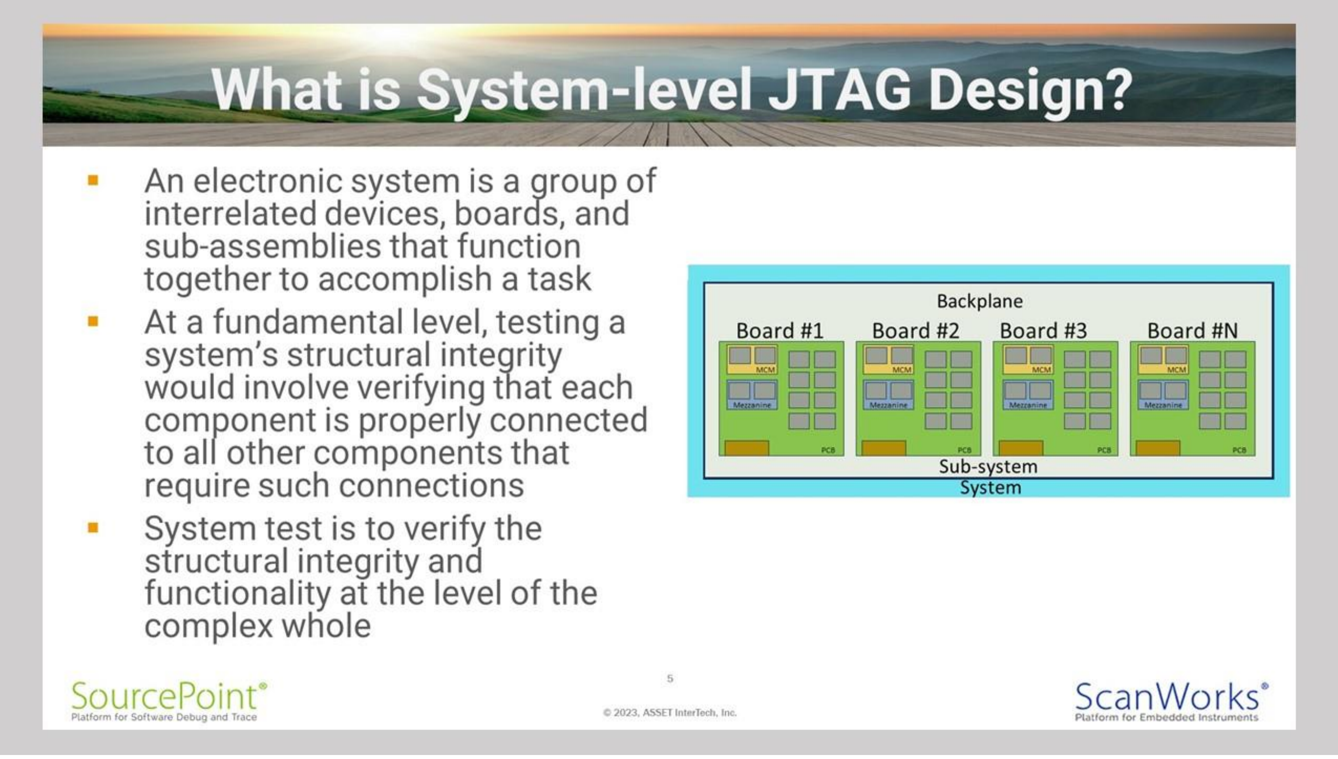



**Figure 4: What is System-level JTAG Design?**

A system has more than one board or subassembly connected. This could be a system assembly as simple as a motherboard with a daughtercard or mezzanine, a multi-chip module or MCM, or a complex computer with hundreds of boards inserted into a backplane. Further, an electronic system is a group of interrelated devices, boards and subassemblies that function together to accomplish a task.

At a fundamental level, testing a system's structural integrity would involve verifying that each component is properly connected to all other components that require such connections. Although the components that comprise an electronic system may be tested individually without regard to the task of the entire system, the overarching goal of system-level testing is to verify the structural integrity and functionality at the level of the complex whole (Figure 4).

## What is System-level JTAG Design - SJTAG

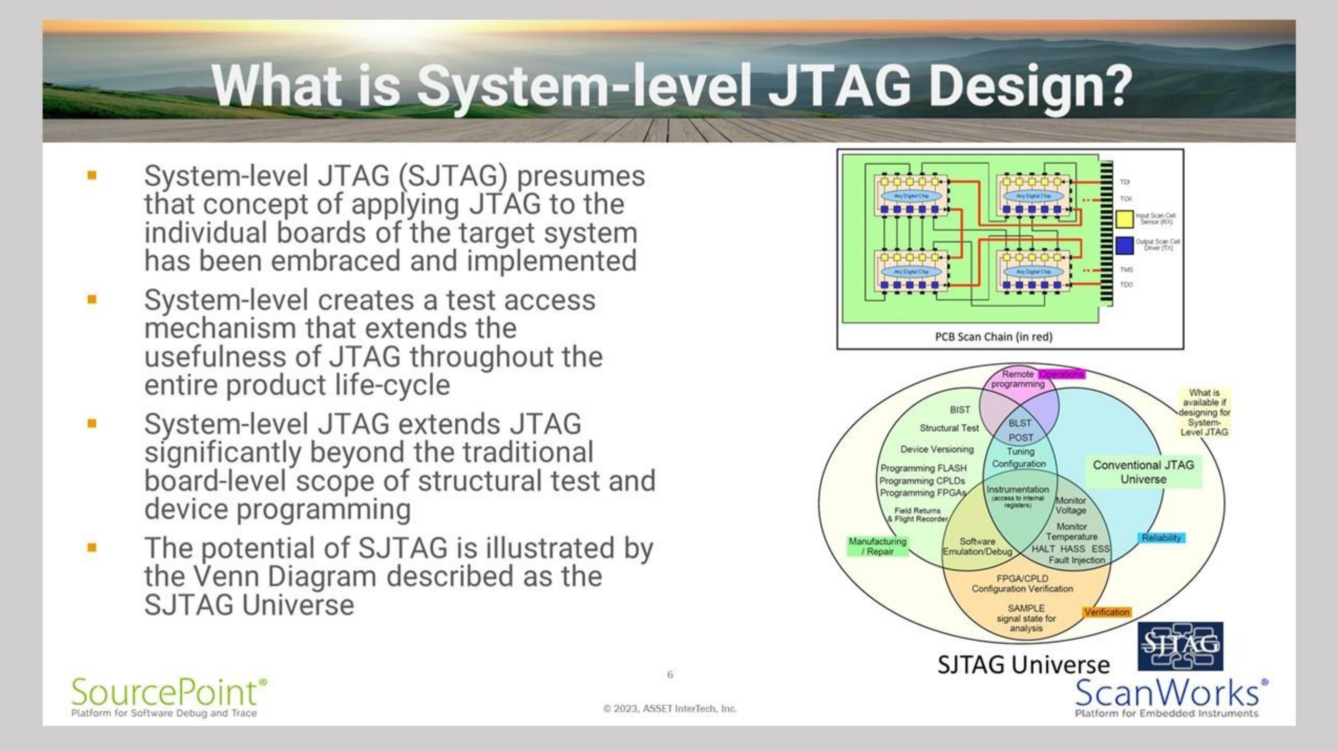



**Figure 5: What is System-level JTAG Design – SJTAG**

System-level JTAG, or SJTAG, expands the application potential of JTAG significantly beyond the traditional board-level scope of structural test and device programming. SJTAG is an expansion of JTAG, so there is a presumption that the concept of applying JTAG at the individual board level has been embraced and implemented.

By taking account of the factors that arise when creating an assembly of boards, and by applying SJTAG design principles to address those factors, the underlying board-level JTAG feature can be leveraged at the system level, creating a test access mechanism that extends the usefulness of JTAG throughout the entire product life cycle. This is illustrated in the Venn Diagram (Figure 5) created by the IEEE SJTAG working group. As you can see, the SJTAG Universe encompasses more than conventional JTAG board testing.

Once you begin to implement system-level JTAG, you'll likely identify additional application areas not included in the diagram. The application potential of SJTAG extends significantly beyond the traditional board-level scope of structural test and device programming.

## System-level JTAG Design Architectures - Ring

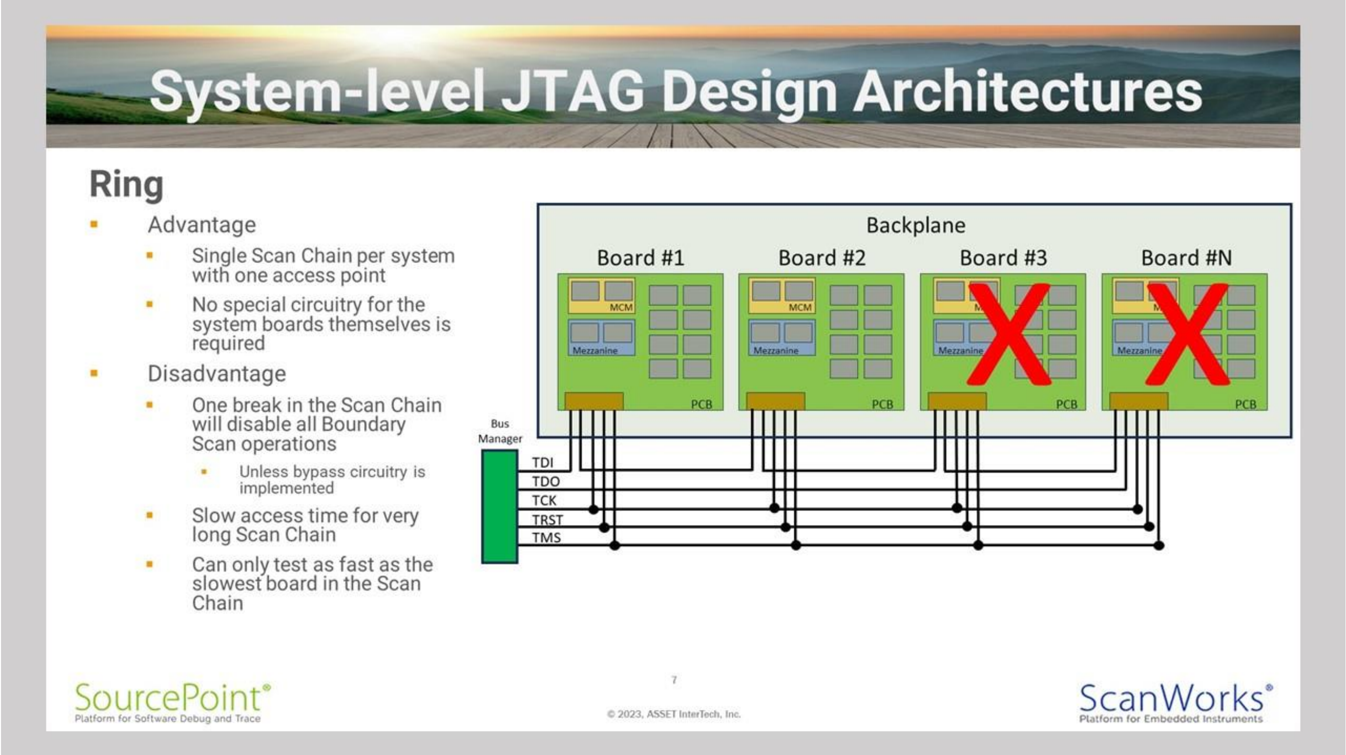



**Figure 6: System-level JTAG Design Architectures – Ring**

The first system-level JTAG architecture we'll examine connects all the boundary scan paths on all the boards in the system to a single scan path with one point of access for the boundary scan test system and a common set of TAP signals. This is known as the Ring architecture.

A single Scan Chain with a single access point is an advantage of this architecture. An additional advantage of this implementation is the reduction or elimination of any special circuitry for the system boards themselves (Figure 6).

A disadvantage of this method is that it requires that the configuration of the system must remain the same in every assembled product. Also, any break in the single scan path will disable all Boundary Scan test access. This break could be caused by a defective board placed in the system, or one of the common

TAP signals becoming corrupted. However, circuity or special connectors could be implemented on the backplane that allow for routing of the data path signal past non-populated or non-working slots.

One very long scan path can also result in slow access times. This architecture presents several challenges for the system designer; there are naturally imposed speed limitations in the parallel clock tree, which can be mitigated somewhat by additional buffer circuitry. Also, the requirement that the test clock rate be maintained in accordance with the slowest board in the system is also an issue.

## System-level JTAG Design Architectures - Star

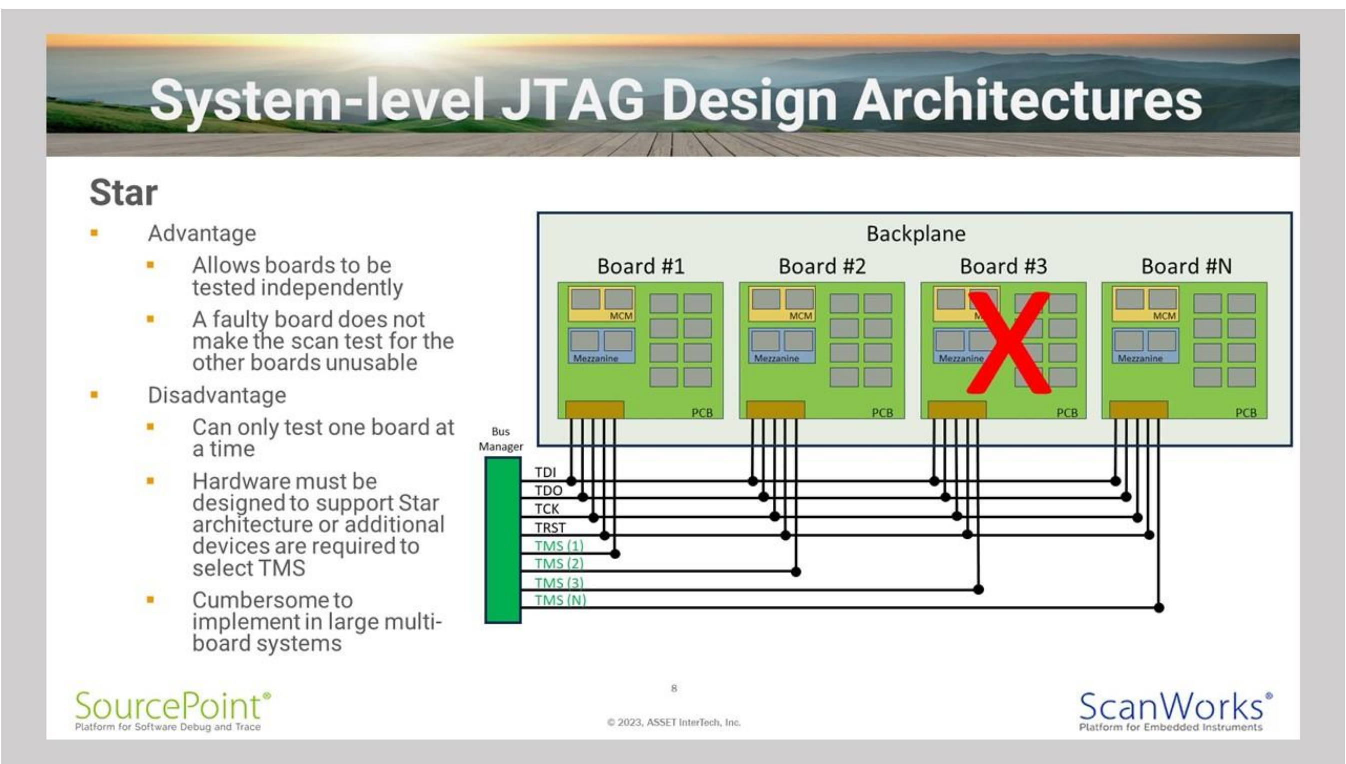



**Figure 7: System-level JTAG Design Architectures – Star**

The second architecture for system-level testing we'll examine is the star architecture. With the Star architecture, the backplane drives TCK and TDI directly to all boards and each board outputs a TDO signal to the backplane (Figure 7).

Each board requires a unique TMS signal from the backplane and a separate circuit trace. During testing, only one board is enabled at a time and only the TMS signal for that board is active. With this architecture, all boards can be tested independently. Also, a faulty board does not make the Boundary

Scan test for the other boards unusable. The test manager interface hardware could be designed to allow for multiple test clock speeds, thereby alleviating the requirement of limiting the test clock speed to be compatible with the slowest board contained in the system.

Disadvantages of this architecture is that only one board can be tested at a time, the bus manager hardware must be designed to support the star architecture or additional devices are required to select TMS and for large multiple-board system the Star architecture is cumbersome to implement.

## System-level JTAG Design Architectures – Multi-drop Device

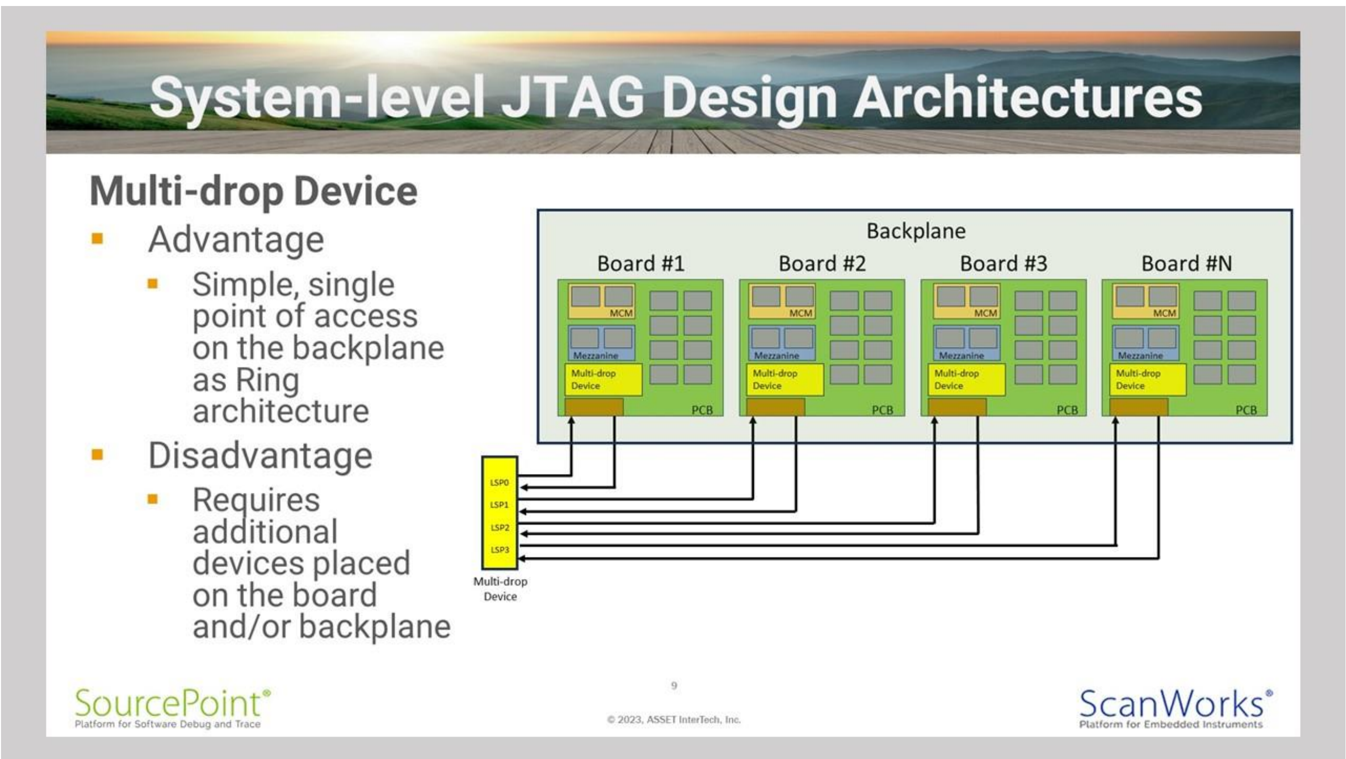



**Figure 8: System-level JTAG Design Architectures – Multi-drop Device**

The next system-level JTAG design architecture we'll examine is implementing a system-level JTAG solution using multi-drop devices. Multi-drop devices such as Scan Path Linkers, Addressable Scan Ports, and Scan Bridges create multiple Scan Paths from a single input TAP source (Figure 8).

Multi-drop systems require minimal system backplane routing resources, and place minimal restrictions, most notably on drive requirements due to the parallel signal connection nature, on the test manager

hardware interface. Using multi-drop devices has the advantage of a single point of access on the backplane.

A disadvantage of using multi-drop devices to implement system-level JTAG is that it does complicate the implementation on the various system-level boards. The system-level boards must be designed to support the multi-drop device. A multi-drop device could also be added to the backplane for additional test flexibility. Clocking and control bus speed limitations are also present when using the multi-drop architecture, to the extent that all multi-drop devices must be in communications and under the control of the test manager to allow for proper system control and synchronization between various system cards. Path selection for the backplane multi-drop device and multi-drop devices on each board can be invoked through a ScanWorks macro.

## System-level JTAG Design Architectures – SCANSTA112

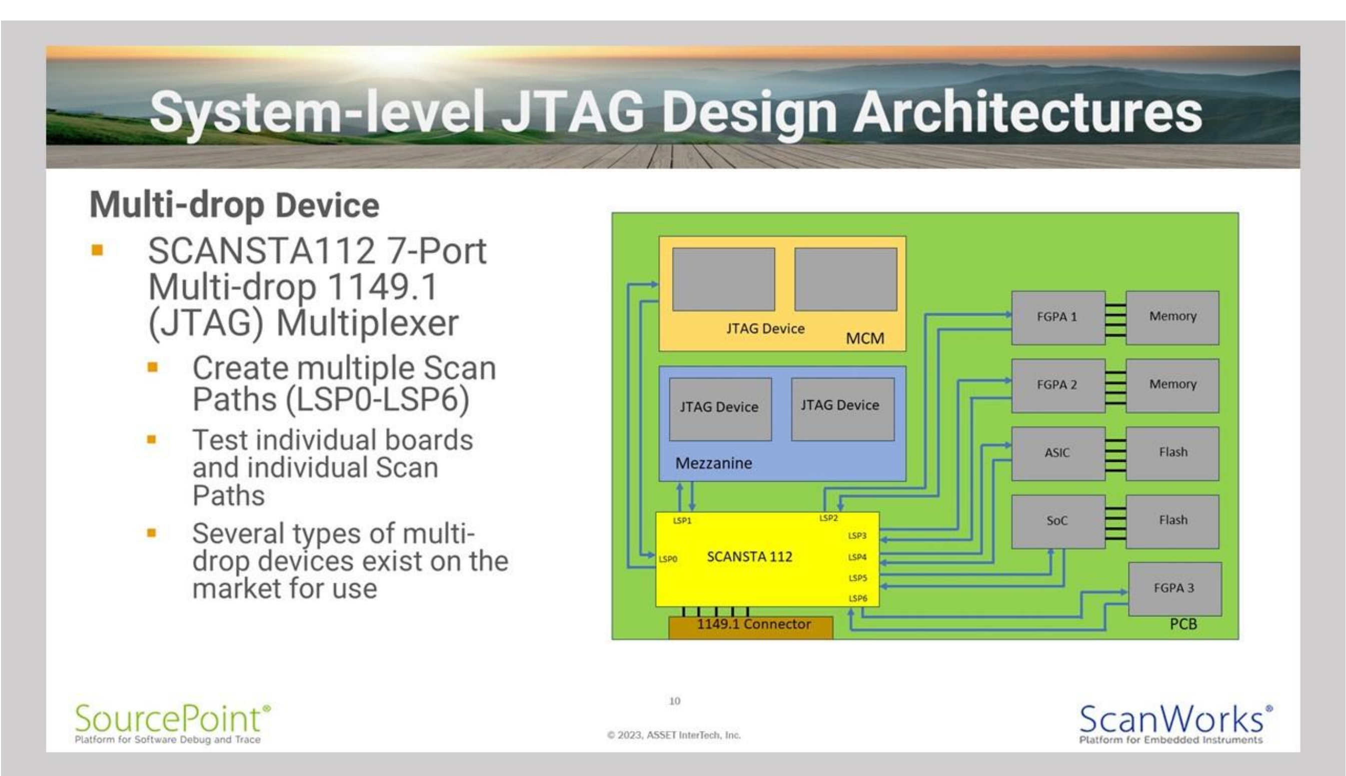


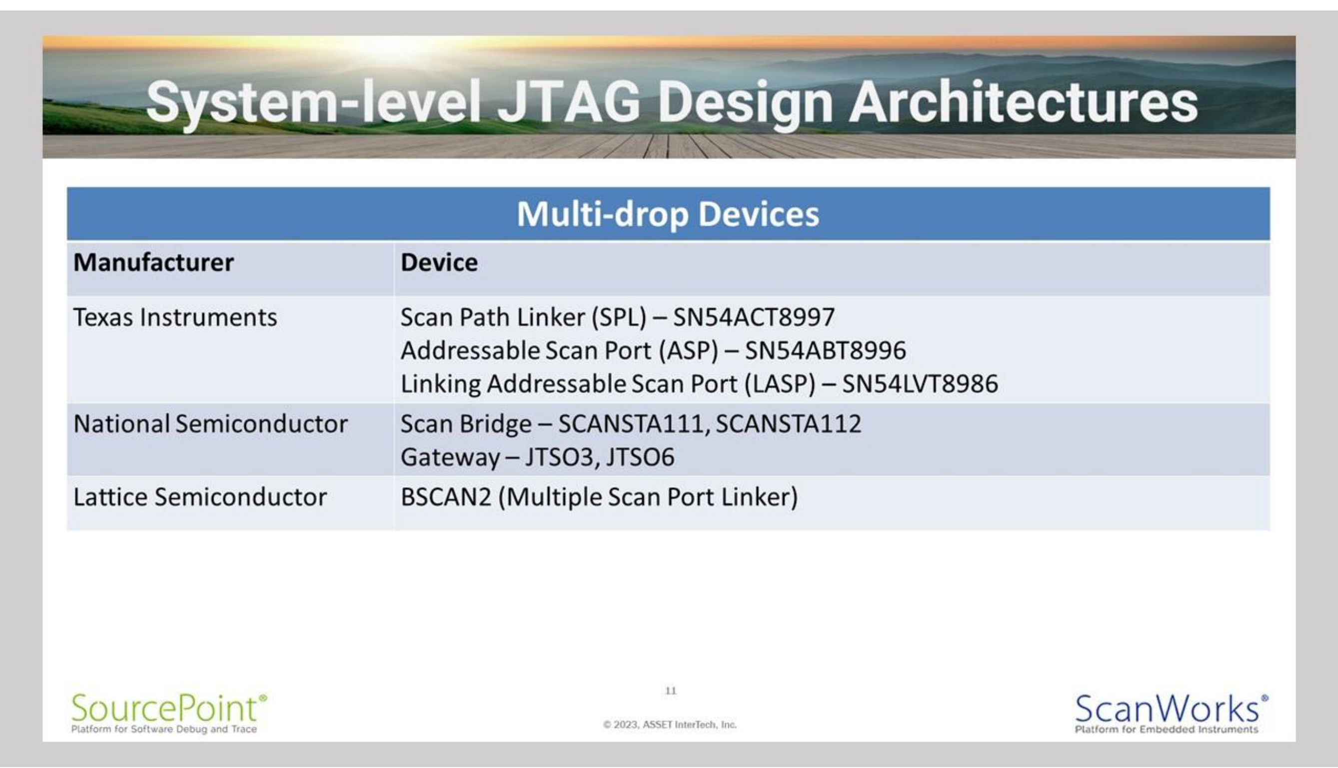

**Figure 9: System-level JTAG Design Architectures – SCANSTA112**

As an example of a multi-drop device that is used to implement system-level JTAG, let's look at the SCANSTA112 7-Port Multi-drop 1149.1 (JTAG) Multiplexer. The SCANSTA112 7-Port Multi-drop

1149.1 (JTAG) Multiplexer creates multiple Scan Chains on the board from a single set of TAP signals. This may be necessary to place Boundary Scan devices on individual Scan Chains for isolation purposes (Figure 9).

For example, the Boundary Scan devices could be FPGAs, or the Boundary Scan devices could have a different TAP voltage than other Boundary Scan devices on the board and require voltage translation. There are several types of multi-drop devices that exist on the market for use.

## System-level JTAG Design Architectures – Multi-drop Device Summary



System-level JTAG Design Architectures

| Multi-drop Devices | |
|---|---|
| **Manufacturer** | **Device** |
| Texas Instruments | Scan Path Linker (SPL) – SN54ACT8997<br>Addressable Scan Port (ASP) – SN54ABT8996<br>Linking Addressable Scan Port (LASP) – SN54LVT8986 |
| National Semiconductor | Scan Bridge – SCANSTA111, SCANSTA112<br>Gateway – JTSO3, JTSO6 |
| Lattice Semiconductor | BSCAN2 (Multiple Scan Port Linker) |



**Figure 10: System-level JTAG Design Architectures – Multi-drop Device Summary**

Here is a list of multi-drop devices that can be used to implement system-level JTAG. You should check the manufacturer data sheet of each device for proper implementation instructions (Figure 10).

## System-level JTAG Design Architectures – Multi-TAP Controller

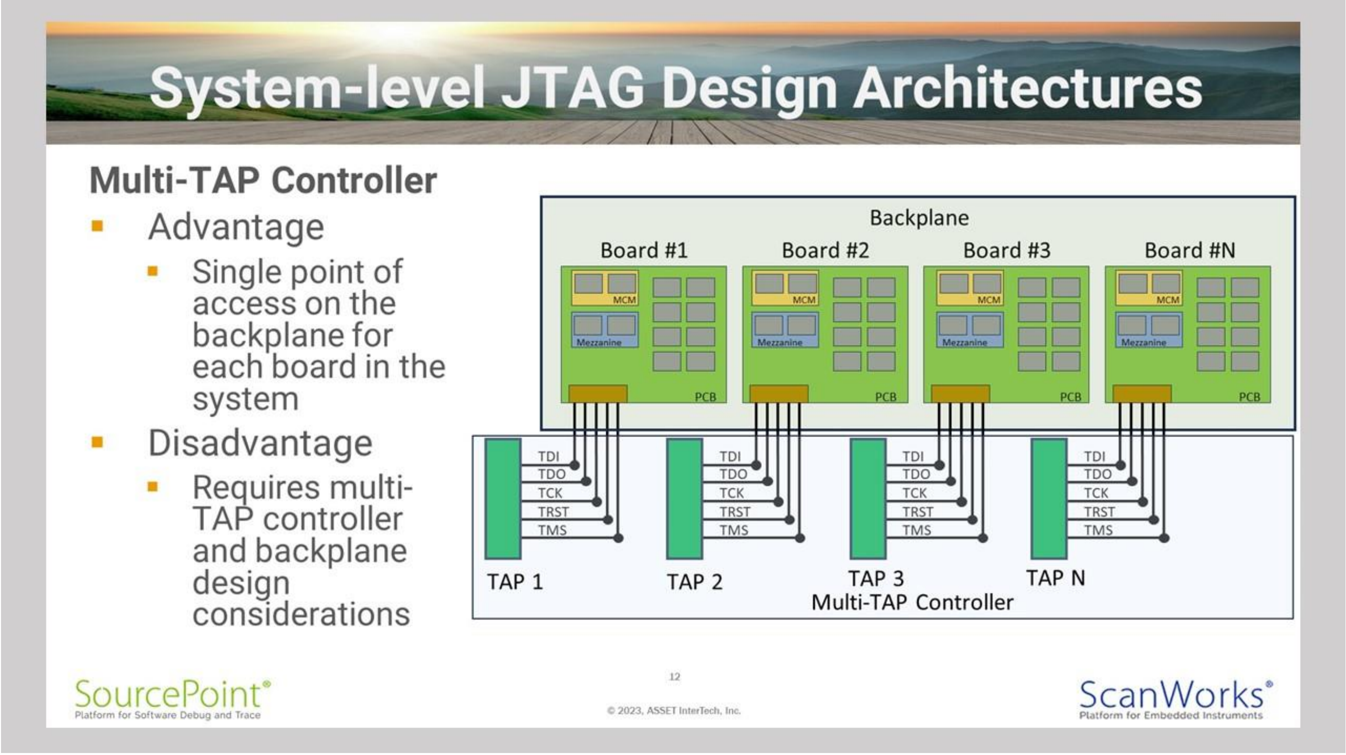



**Figure 11: System-level JTAG Design Architectures – Multi-TAP Controller**

There are also external hardware methods that can be used to implement system-level JTAG. One hardware method uses a multi-TAP controller.

Using a multi-TAP controller has the advantage of single points of access on the backplane for each board in the system. Therefore, each board can be accessed and tested independently. Should one board be faulty or its TAP connections corrupted, the test for that board within the system can be bypassed (Figure 11).

Using an ASSET multi-TAP controller, along with our ScanWorks software, board tests can be applied individually or in a concatenated fashion, meaning the boards in the system can be tested as one very long Scan Chain. This external method of system-level JTAG does require the acquisition of multi-TAP hardware at added cost, and the backplane must be designed to accommodate the multiple TAPs from the controller.

## System-level JTAG Design Architectures – PCIe-410

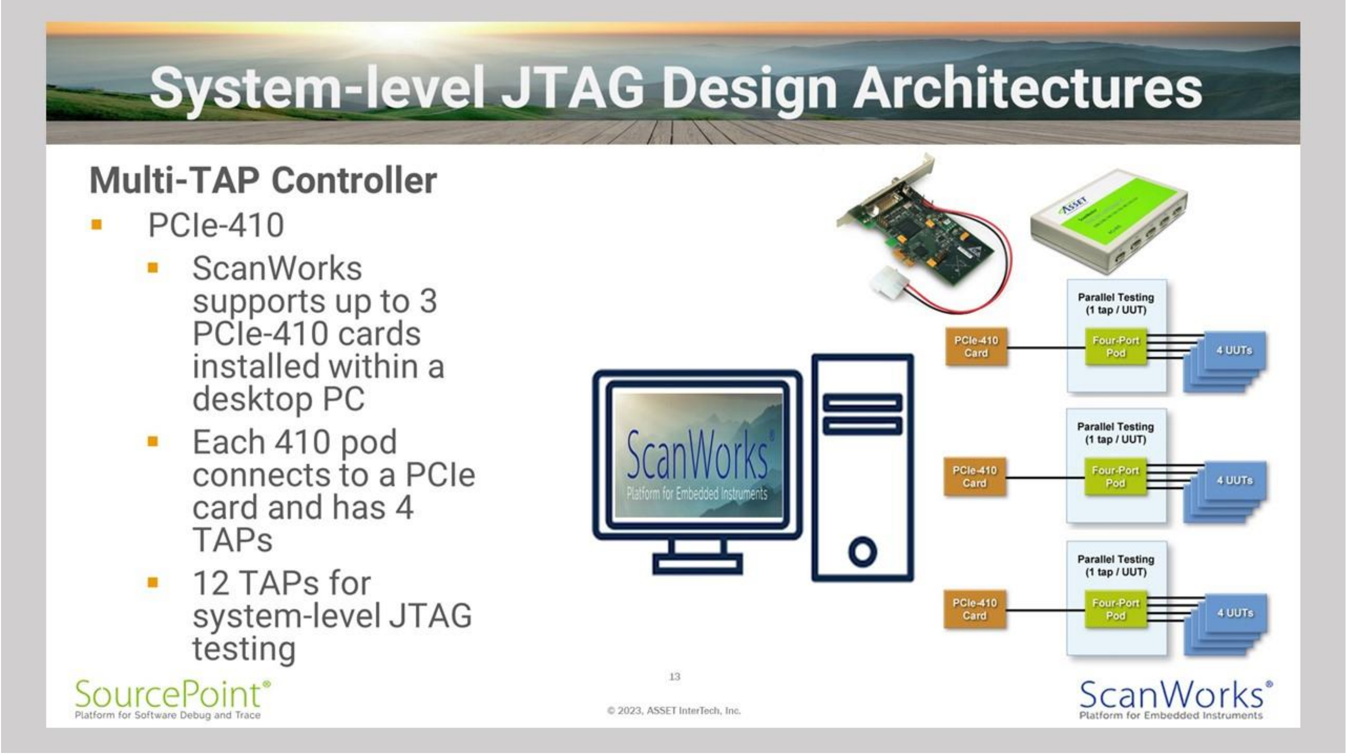
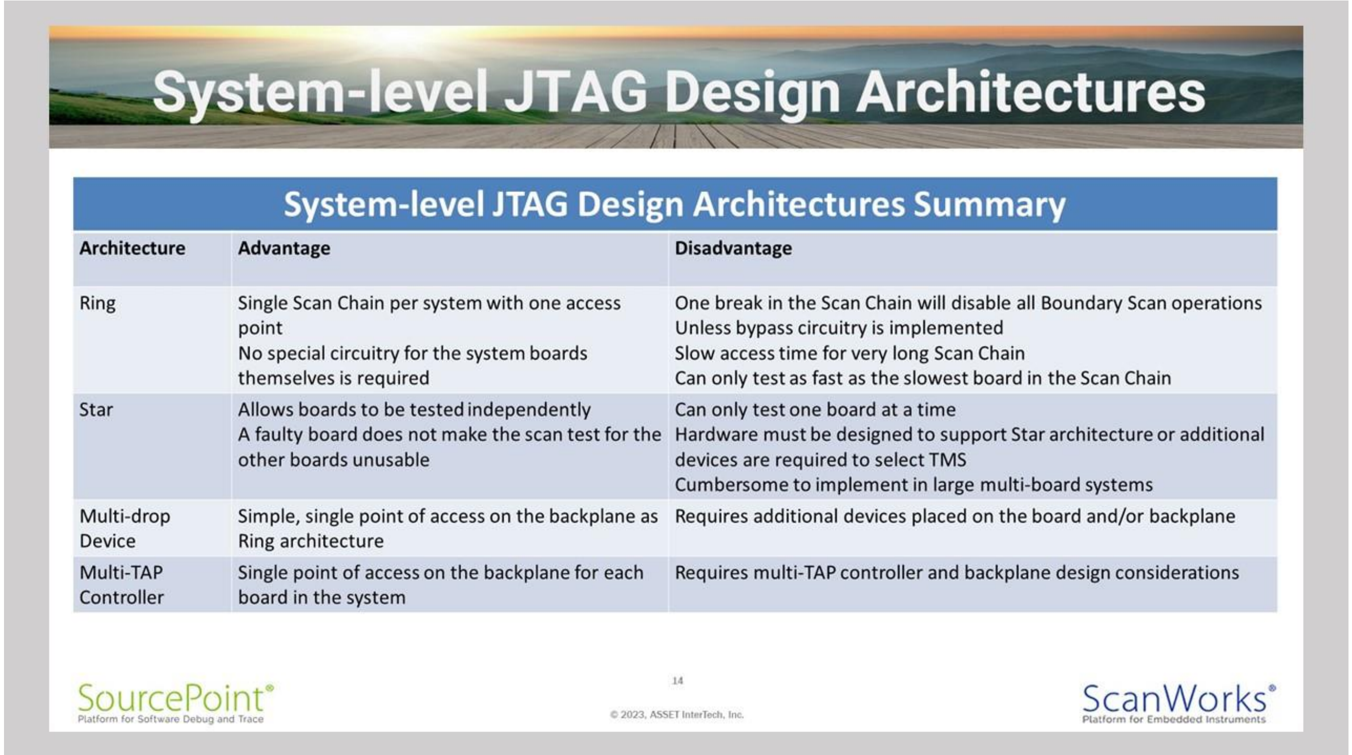



**Figure 12: System-level JTAG Design Architectures – PCIe-410**

An example of using a multi-TAP controller to implement a system-level JTAG solution is ASSET's PCIe-410 hardware controller. ScanWorks supports up to 3 PCIe cards installed within a desktop PC. Each ASSET 410 pod connects to a PCIe card and has 4 TAPs (Figure 12).

With proper backplane design, this configuration provides up to 12 TAPs for individual board-level or system-level JTAG testing. Or by using the ScanWorks concatenation feature, the printed circuit boards could be tested as one very large Scan Chain. Combining a multi-TAP controller along with placing multi-TAP devices on a backplane can create even more system-level JTAG opportunities.

## System-level JTAG Design Architectures – Summary




System-level JTAG Design Architectures Summary

| Architecture | Advantage | Disadvantage |
|---|---|---|
| Ring | Single Scan Chain per system with one access point<br>No special circuitry for the system boards themselves is required | One break in the Scan Chain will disable all Boundary Scan operations Unless bypass circuitry is implemented<br>Slow access time for very long Scan Chain<br>Can only test as fast as the slowest board in the Scan Chain |
| Star | Allows boards to be tested independently<br>A faulty board does not make the scan test for the other boards unusable | Can only test one board at a time<br>Hardware must be designed to support Star architecture or additional devices are required to select TMS<br>Cumbersome to implement in large multi-board systems |
| Multi-drop Device | Simple, single point of access on the backplane as Ring architecture | Requires additional devices placed on the board and/or backplane |
| Multi-TAP Controller | Single point of access on the backplane for each board in the system | Requires multi-TAP controller and backplane design considerations |

**Figure 13: System-level JTAG Design Architectures – Summary**

As a summary, here are the system-level architectures presented along with the advantages and disadvantages of each approach (Figure 13).

## System-level JTAG Design – Embedded Applications – SED for Test

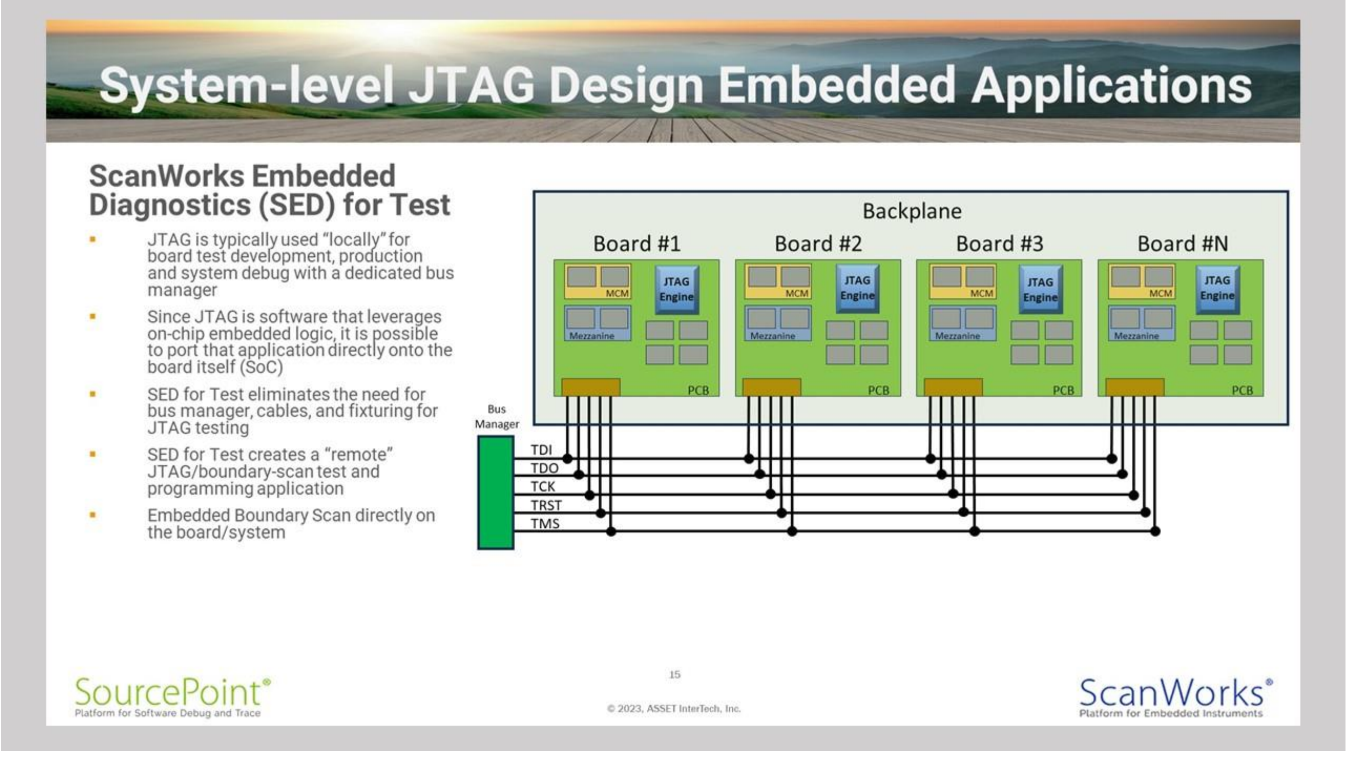



**Figure 14: System-level JTAG Design – Embedded Applications – SED for Test**

JTAG is typically used "locally" for board test development, production, and system debug with a dedicated bus manager. Since JTAG is software that leverages on-chip embedded logic, it is possible to port that application directly onto the board itself. SED for Test eliminates the need for the bus manager, cables, and fixturing for JTAG testing. SED for Test creates a "remote" JTAG/boundary-scan test and programming application. Essentially with SED for Test, Boundary Scan is embedded directly onto the printed circuit board (Figure 14).

## System-level JTAG Design – Embedded Applications – SED for Test Advantages

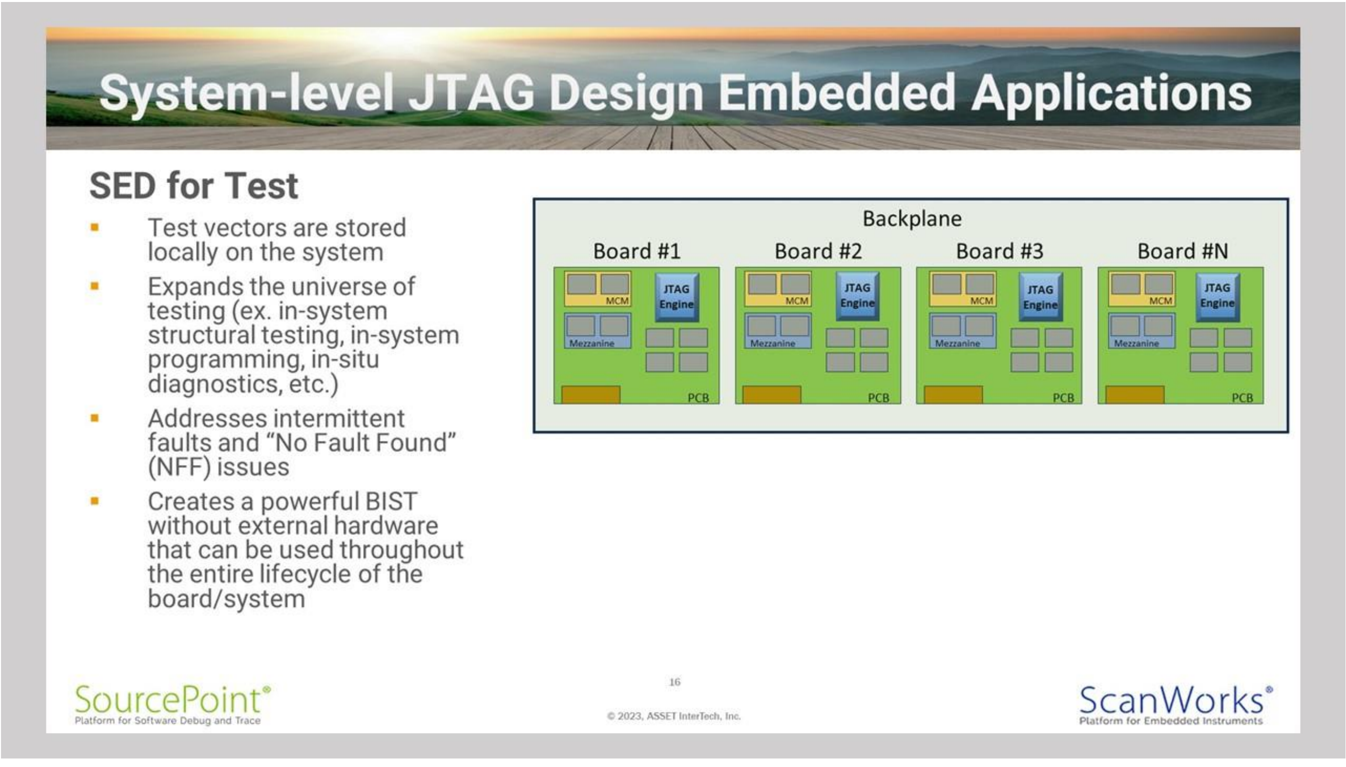



**Figure 15: System-level JTAG Design – Embedded Applications – SED for Test Advantages**

SED for Test has attributes that make it attractive for implementation. The test vectors are stored locally on the system. SED for Test expands the universe of testing (ex. in-system structural testing, in-system programming, in-situ diagnostics, etc.). SED for Test addresses intermittent faults and "No Fault Found" (NFF) issues. SED for Test creates a powerful BIST without external hardware that can be used throughout the entire lifecycle of the printed circuit board/system (Figure 15).

If you'd like more information on SED for Test, I conducted a webinar describing it in detail along with a demonstration of how the SED application is embedded within an SoC. This webinar recording is on the ASSET website.

## System-level JTAG Design – Embedded Applications – SED for Test Resources

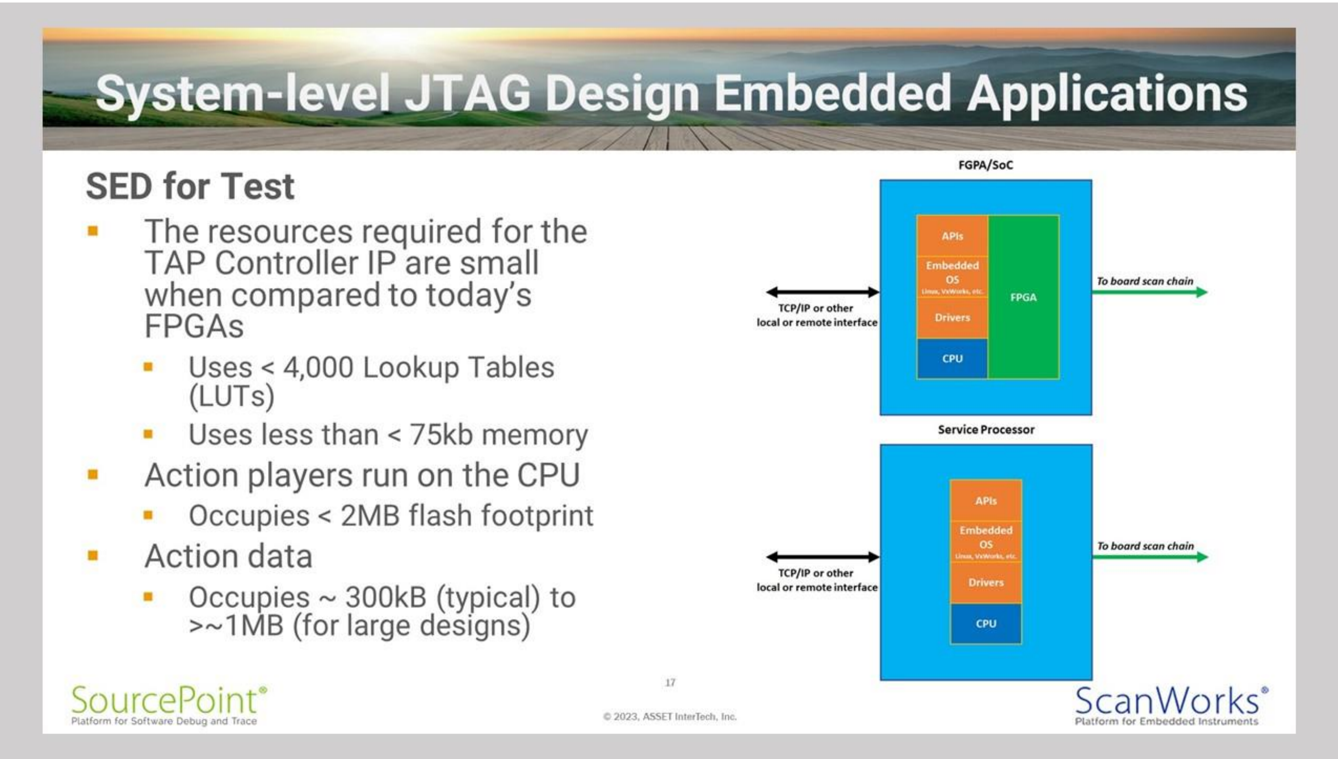



**Figure 16: System-level JTAG Design – Embedded Applications – SED for Test Resources**

A common question regarding SED for Test is "how much real estate will an SED for Test implementation require in my FPGA or processor?"

- The resources required for the TAP Controller IP are small when compared to today's FPGA's
  - Uses < 4,000 Lookup Tables (LUTs)
  - Uses < 75kb memory
- Action players run on the CPU
  - Occupies <2MB flash footprint
- Acton data occupies ~300 kB for typical designs and >~1MB for large designs (Figure 16)

## System-level JTAG Design – Embedded Applications – SED for Debug

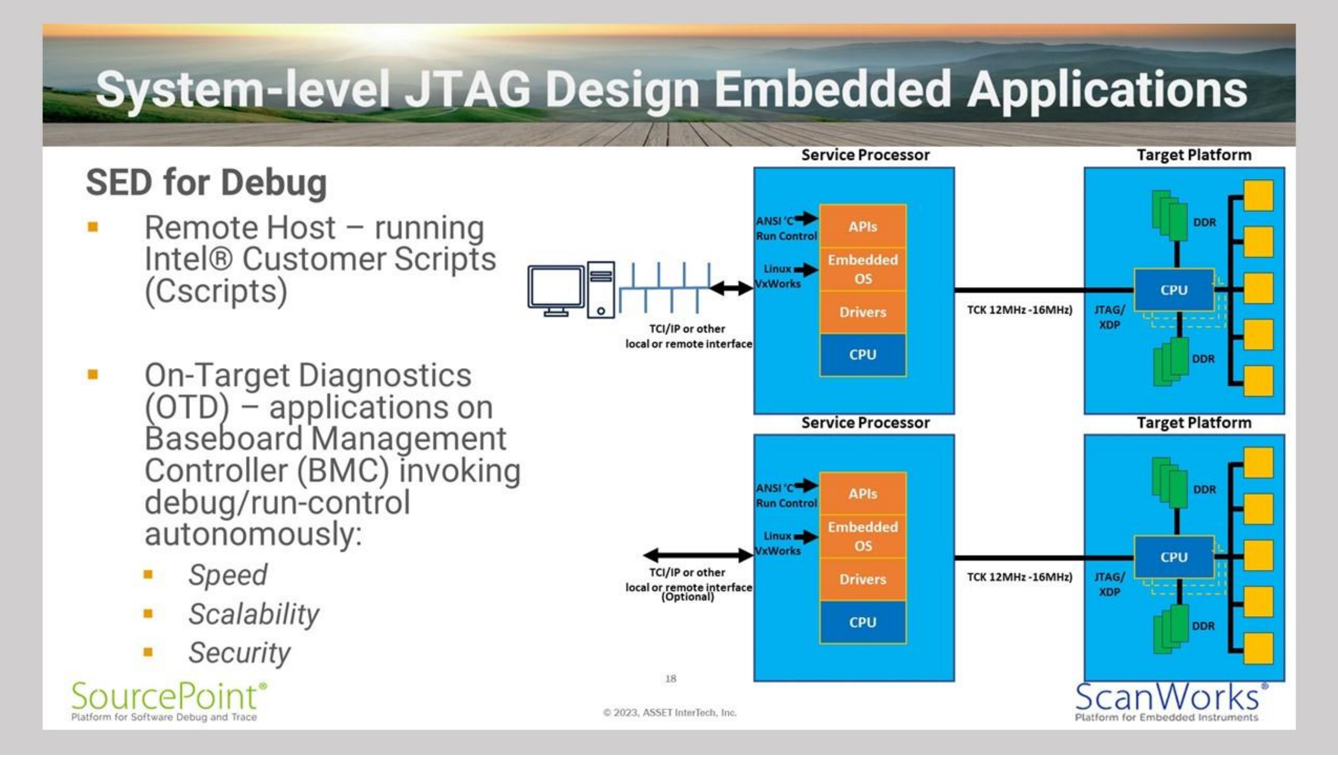



**Figure 17: System-level JTAG Design – Embedded Applications – SED for Debug**

System-Level JTAG also applies to debug applications. In this instance, a service processor (also known as Baseboard Management Controller – BMC - in the server world) can host the JTAG Mastering function and a library of debugging primitives to read and write registers, memory, and IO. This can operate by running the Intel Customer Scripts (Cscripts) on a remote host, or alternatively at-scale via an On-Target Diagnostic. These OTDs can run on hundreds or even thousands of individual server nodes to provide in-situ, out-of-band debugging services (Figure 17).

## System-level JTAG Design Embedded Applications – SED for Built-in Self-Test

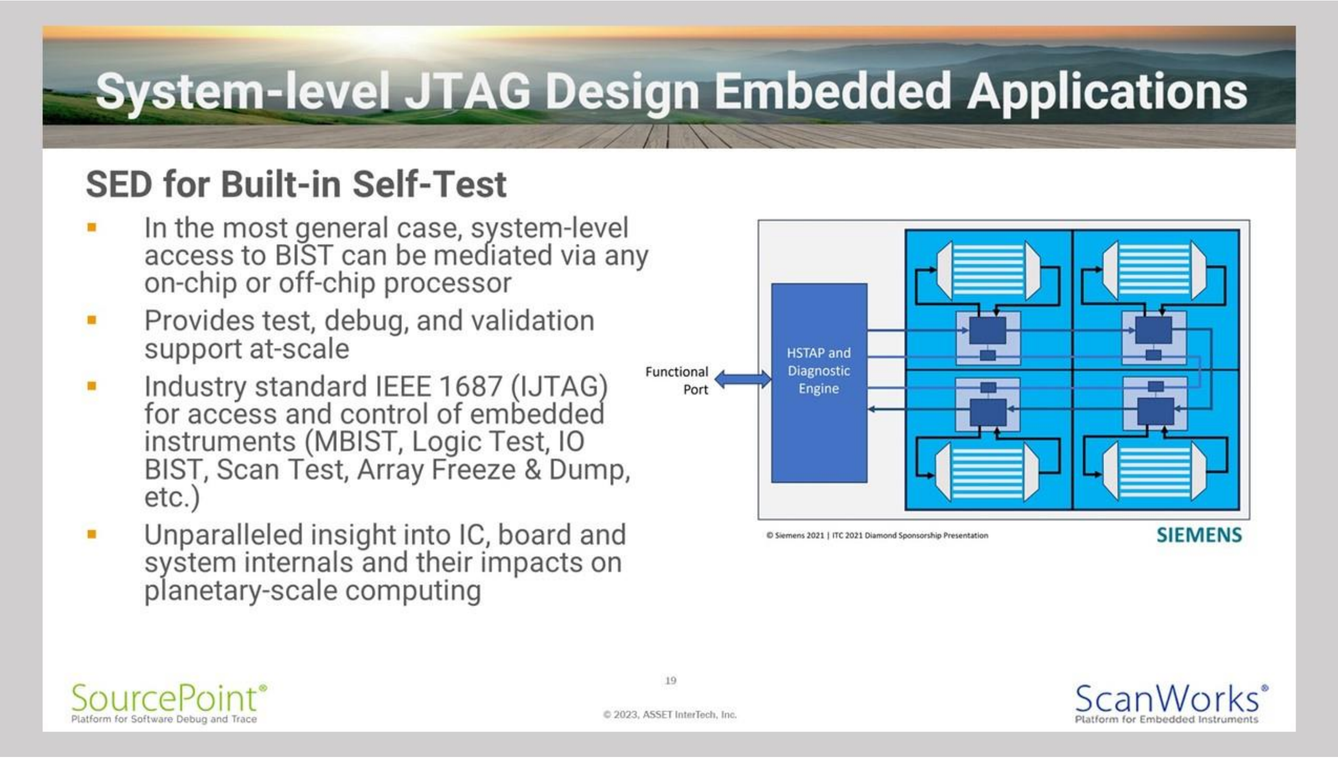



**Figure 18: System-level JTAG Design – Embedded Applications – SED for Built-in Self-Test**

In the most generic sense, system-level JTAG provides at-scale access to any on-chip Built-In Self-Test, or BIST. This would include Memory BIST, Logic Test, Scan Test, and many other services that can be accessed and controlled via internal IJTAG networks within ICs. This is an example image, courtesy of Siemens EDA, that shows a Streaming Scan Network (SSN) within an IC that can be accessed via JTAG or a High-Speed TAP to stream scan and memory data out of a device. This has potential for system-level, in-field testing of devices in-situ, providing an unparalleled level of diagnostic granularity for extremely intermittent issues, such as Silent Data Errors (Figure 18).

## Guidelines for System-level JTAG Design – Summary

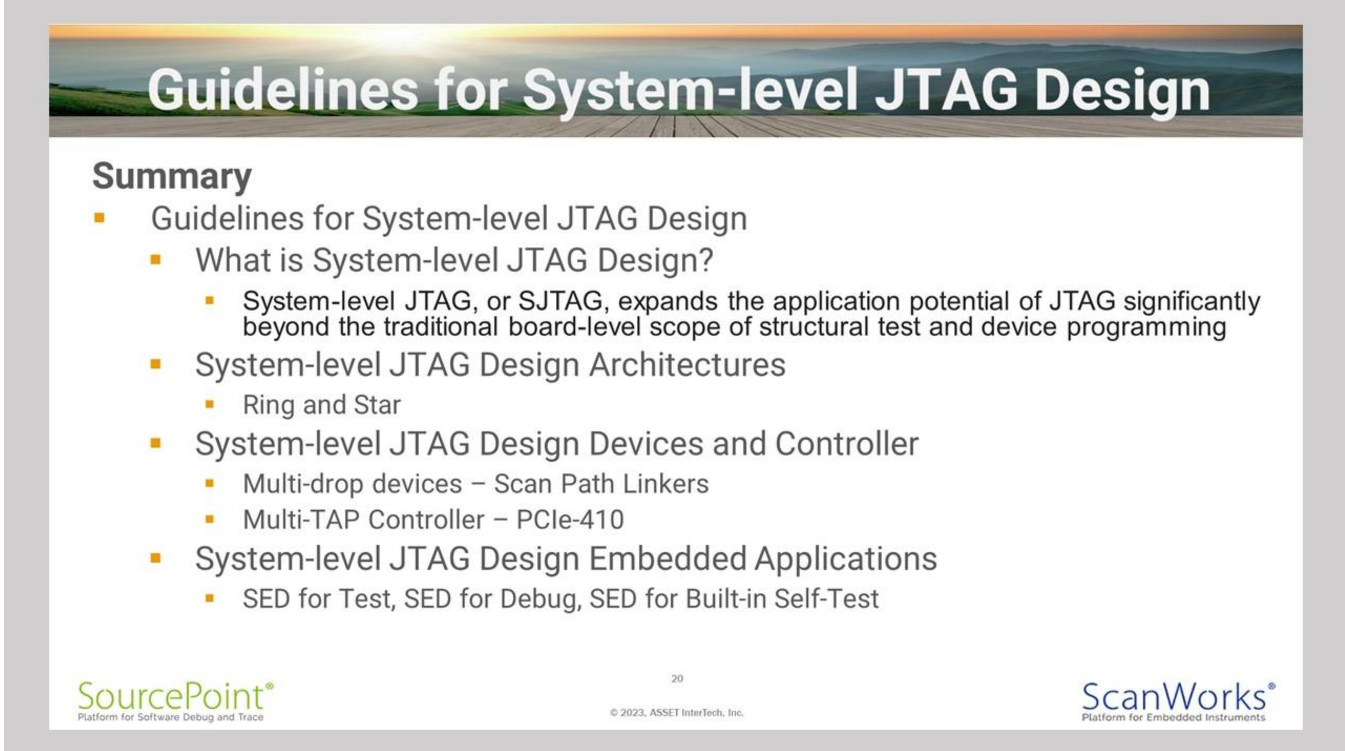



**Figure 19: Guidelines for System-level JTAG Design – Summary**

So, in summary. These are the guidelines for system-level JTAG design we've covered in this eBook (Figure 19).

## System-level JTAG Desing – Additional Resources

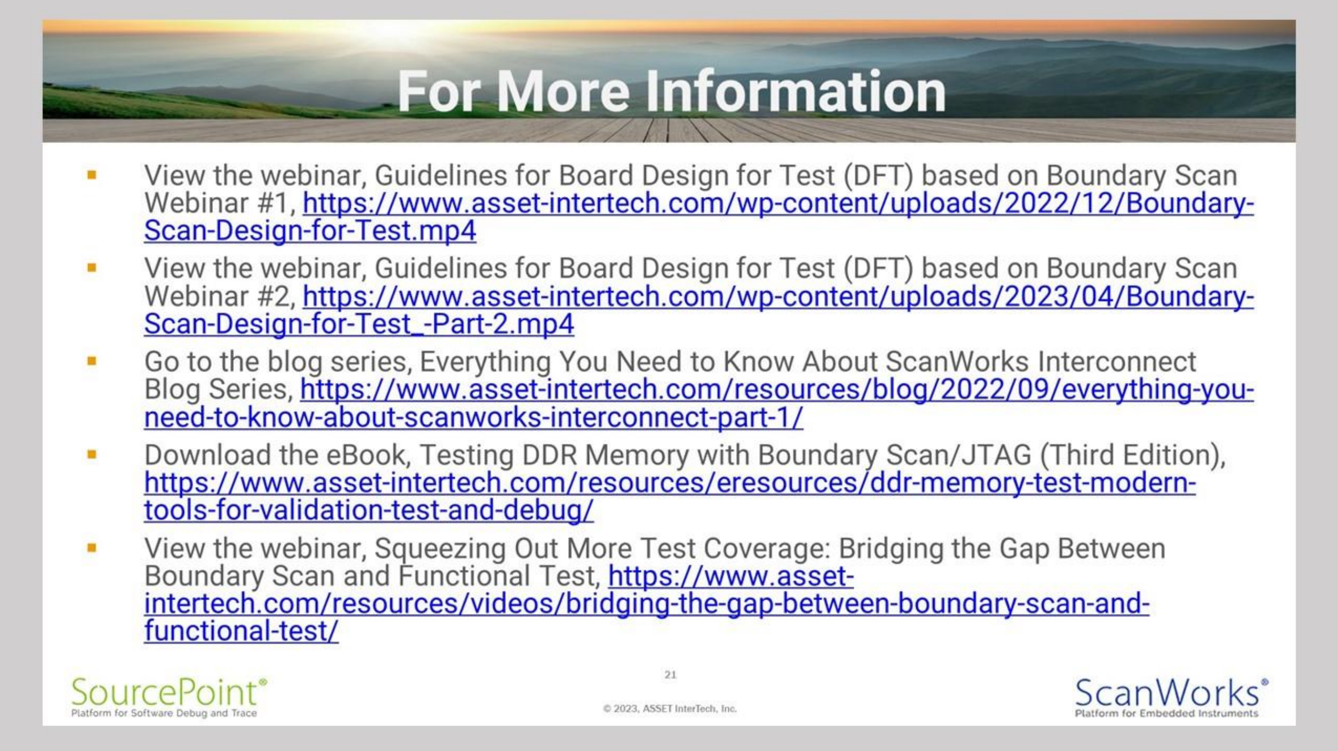



**Figure 20: System-level JTAG Design – Additional Resources**

There are also more resources that you can review on our website on a variety of test-related topics and the use of the ScanWorks Boundary Scan Test tool (Figure 20).

## Conclusion

In this eBook, we examined guidelines for system-level JTAG design. System-level JTAG testing ensures the boards within a system are structurally sound. System-level JTAG also ensures complex system functions as designed when assembled. Testing boards individually with JTAG, as they are produced, is valuable. Detecting structural faults, device programming, and functional testing of boards before system assembly is beneficial to ensuring proper system functionality. The overarching goal is for the complete system to function properly, system-level JTAG ensures this. Detecting and repairing faults on individual boards is key to reducing overall production cost. Several types of faults can take place when the entire system is assembled. Implementing system-level JTAG can be quite useful in testing backplane connections, connectors, backplane passive and active circuitry. Testing a complex system can also be used during functional testing and while the entire system is undergoing environmental testing as well.

There are several architecture designs which make system-level JTAG possible. Design for Test (DFT) considerations of each design approach must be given considered before implementation. Aside from the physical layout considerations, each architecture design has implementation advantages and disadvantages. Multi-drop devices such as Scan Path Linkers, Addressable scan ports, and Scan Bridges can also be used to create multiple Scan Paths on a backplane. Multiple Scan Paths on a backplane can create a very flexible system-level JTAG infrastructure. Multi-TAP hardware can also be used to implement system-level JTAG.

Embedded applications can also be used to implement a system-level JTAG solution. Since JTAG is software that leverages on-chip embedded logic, it is possible to port that application directly onto the board itself. SED for Test eliminates the need for the bus manager, cables, and fixturing for JTAG testing. SED for Test creates a "remote" JTAG/boundary-scan test and programming application.

System-level JTAG can be used throughout the life cycle of the product. Starting with design, manufacturing, and eventually testing complete installed systems within installed products, system-level JTAG can play an important role. Maximizing board test coverage and system level test functionality is imperative to improving manufacturing yields, increasing product quality, and reducing product returns. Ultimately, Boundary Scan system test ensures a high-quality system.