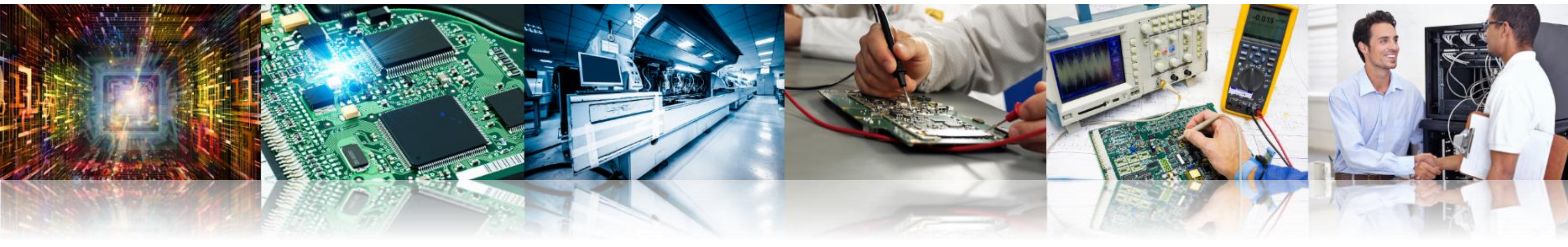
JTAG Tutorial

The IEEE-1149.1 standard, also known as JTAG or boundary-scan, has for many years provided an access method for testing printed circuit board assemblies, in-system-programming, and more. But what is JTAG, and how can it be used to benefit organizations in diverse industries across all phases of the product life cycle?

CORELIS An EWA Company



Corelis, Inc. 13100 Alondra Blvd. Cerritos, CA 90703-2146 (562) 926-6727

www.corelis.com

What is JTAG?

Introduction

Since its introduction as an industry standard in 1990, JTAG has continuously grown in adoption, popularity, and usefulness—even today, new revisions and supplements to the IEEE-1149.1 standard are being developed and implemented. This document is a brief introduction to the nature and history of JTAG, from its introduction to new extensions in current development.

What is JTAG?

JTAG, commonly referred to as boundary-scan and defined by the Institute of Electrical and Electronic Engineers (IEEE) 1149.1, originally began as an integrated method for testing interconnects on printed circuit boards (PCBs) implemented at the integrated circuit (IC) level. As PCBs grew in complexity and density—a trend that continues today—limitations in the traditional test methods of in-circuit testers (ICTs) and bed of nails fixtures became evident. Packaging formats, specifically Ball Grid Array (BGA, depicted in Figure 1) and other fine pitch components, designed to meet ever-increasing physical space constraints, also led to a loss of physical access to signals.

These new technology developments led to dramatic increases in costs related to designing and building bed of nails fixtures; at the same time, circuit board test coverage also suffered. JTAG/boundary-scan presented an elegant solution to this problem: build functionality into the IC to assist in testing assembled electronic systems.

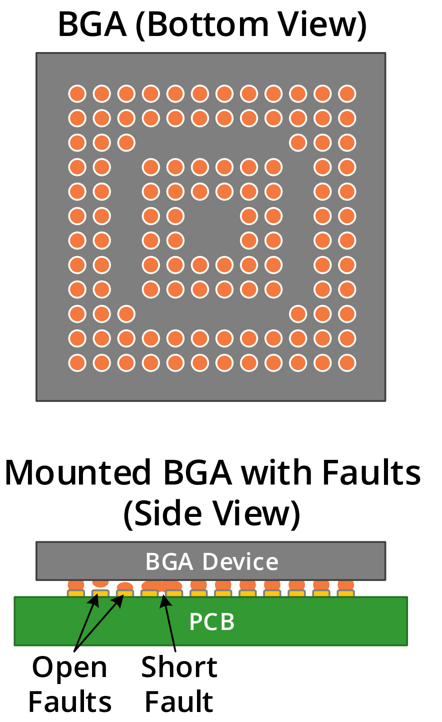


Figure 1. BGA faults are difficult to detect and diagnose without JTAG.

Today, JTAG is used for everything from testing interconnects and functionality on ICs to programming flash memory of systems deployed in the field and everything in-between. JTAG and its related standards have been and will continue to be extended to address additional challenges in electronic test and manufacturing, including test of 3D ICs and complex, hierarchical systems.

History of JTAG

In the 1980s, the Joint Test Action Group (JTAG) set out to develop a specification for boundary-scan testing that was standardized in 1990 as the IEEE Std. 1149.1-1990. A few years later in 1993, a new revision to the standard—1149.1a—was introduced to clarify, correct, and enhance the original specification. An additional supplement, 1149.1b, was published in 1994 to add Boundary-Scan Description Language (BSDL) to the standard, paving the way for fast, automated test development and spurring continuous adoption by major electronics producers all over the world. The lessons that were learned became formalized in an update to the core standard in 2001 and IEEE-1149.1-2001 was published.

As new applications of JTAG were discovered, new standards were developed to extend the capabilities of JTAG. Standards such as the IEEE-1149.5 module test and maintenance bus standard in 1995 and the IEEE-1149.4 standard for mixed-signal testing in 1999 were met with low adoption rates and are not widely used at present. The IEEE-1149.6 standard introduced in 2003, on the other hand, began with slow adoption but has since become standard in many ICs as the technology it addressed—high-speed, AC-coupled signals—became a common feature of electronic systems. IEEE-1149.7, published in 2009 to address the need for JTAG in low-pin-count systems, is now standard on many popular microcontrollers.

Additional standards have also been published to add specific test capabilities. In 2002, the IEEE-1532 standard for in-system configuration of programmable devices was released and is now a common feature of FPGAs and their supporting software systems. IEEE-1581 was developed in 2011 to provide a convenient method of testing interconnects of high-speed memories with slow-speed test vectors; a version of this capability is implemented in some DDR4 memory components. To address the new application of combined capacitive sensing and boundary-scan test, IEEE-1149.8.1 was published in 2012. The extensibility of JTAG has been proven time and again.

Timeline of JTAG-related Standards

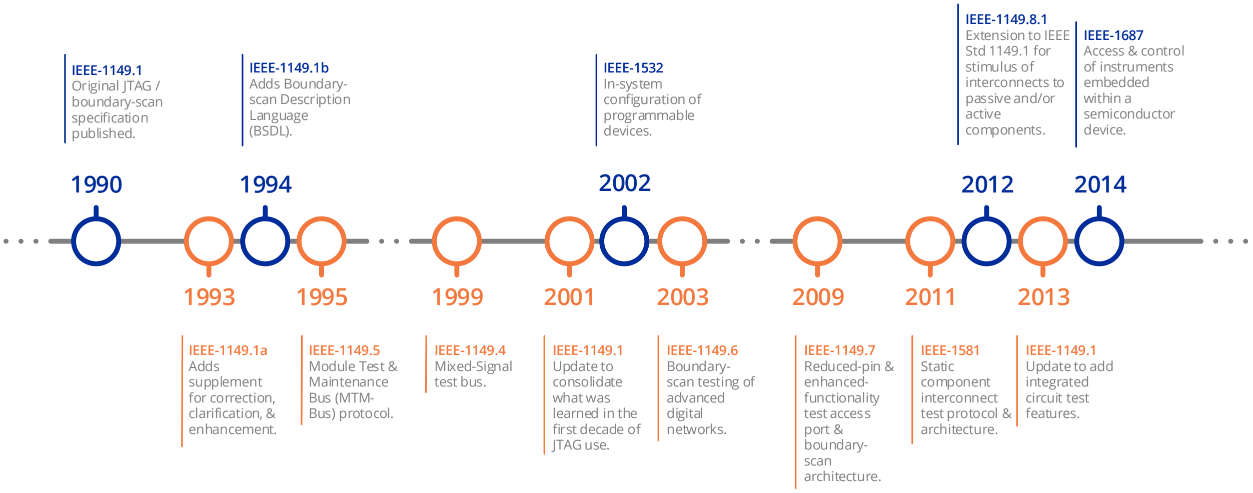

IEEE-1149.1
Original JTAG / boundary-scan specification published.
IEEE-1149.1b
Adds Boundary-scan Description Language (BSDL).
IEEE-1532
In-system configuration of programmable devices.
IEEE-1149.8.1
Extension to IEEE Std 1149.1 for stimulus of interconnects to passive and/or active components.
IEEE-1687
Access & control of instruments embedded within a semiconductor device.
1990
1994
2002
2012
2014
1993
1995
1999
2001
2003
2009
2011
2013
IEEE-1149.1a
Adds supplement for correction, clarification, & enhancement.
IEEE-1149.5
Module Test & Maintenance Bus (MTM-Bus) protocol.
IEEE-1149.4
Mixed-Signal test bus.
IEEE-1149.1
Update to consolidate what was learned in the first decade of JTAG use.
IEEE-1149.6
Boundary-scan testing of advanced digital networks.
IEEE-1149.7
Reduced-pin & enhanced-functionality test access port & boundary-scan architecture.
IEEE-1581
Static component interconnect test protocol & architecture.
IEEE-1149.1
Update to add integrated circuit test features.

Figure 2. JTAG has been under continuous development for more than 20 years.

More recently, efforts have been made to standardize JTAG access to instruments embedded within ICs. The IEEE-1149.1 standard was updated once more in 2013 for some housekeeping and to add extensions to access these instruments. Just one year later, an alternative standard for accessing these instruments, IEEE-1687, was published. Looking to the future, industry activities to extend JTAG into 3D-IC testing, system-level testing, and high-speed testing are already underway, proving that the versatility and extensibility of JTAG is here to stay.

How JTAG Works

The JTAG/boundary-scan test architecture was originally developed as a method to test interconnects between ICs mounted on a PCB without using physical test probes. Boundary-scan cells created using multiplexer and latch circuits are attached to each pin on the device. These cells, embedded in the device, can capture data from pin or core logic signals as well as force data onto pins. Captured data is serially shifted out through the JTAG Test Access Port (TAP) and can be compared to expected values to determine a pass or fail result. Forced test data is serially shifted into the boundary-scan cells. All of this is controlled from a serial data path called the scan path or scan chain.

Because each pin can be individually controlled, boundary-scan eliminates a large number of test vectors that would normally needed to properly initialize sequential logic. Using JTAG, tens or hundreds of test vectors may do the job that had previously required thousands. Boundary-scan enables shorter test times, higher test coverage, increased diagnostic capability, and lower capital equipment cost.

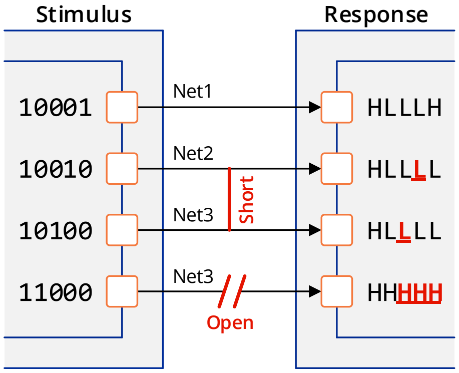


Figure 3. Basic principles of an interconnect test.

The principles of interconnect test using boundary-scan components are illustrated in Figure 3. Two boundary-scan compliant devices are connected with four nets. The first device includes four outputs that are driving the four inputs of the other with predefined values. In this case, we assume that the circuit includes two faults: a short fault between Net2 and Net3, and an open fault on Net4. We will also assume that a short between two nets behaves as a wired-AND and an open fault behaves as a stuck-at-1 condition.

To detect and isolate defects, the tester shifts the patterns shown in Figure 3 into the first boundary-scan register and applies these patterns to the inputs of the second device. The input values captured in the boundary-scan register of the second device are shifted out and compared to the expected values. In this case, the results, underlined and marked in red on Net2, Net3, and Net4, do not match the expected values and the tester tags these nets as faulty. Sophisticated algorithms are used to automatically generate the minimal set of test vectors to detect, isolate, and diagnose faults to specific nets, devices, and pins.

Of course, interconnect testing is just one of many uses of JTAG—the aforementioned JTAG TAP has been extended to support additional capabilities including in-system-programming (ISP), in-circuit-emulation (ICE), embedded functional testing, and many more. The standard accounts for the addition of device-specific instructions and registers that can be used to interact with additional IC capabilities. For example, a microprocessor device may have embedded functionality for data download, program execution, or register peek-and-poke activities accessible using JTAG TAP; using the same tools, FPGA and CPLD devices can be erased, configured, read-back, and controlled using JTAG instructions through the IEEE-1532 standard. More recently, embedded IC instrumentation—from instruments that measure voltage and current to devices that can execute high-speed test on the chip—has used the JTAG TAP as the access mechanism, providing new visibility into the IC and further expanding the scope of JTAG testing.

Product Life-Cycle Phases and Applications

While JTAG/boundary-scan was originally regarded as a method to test electronic products during the production phase, new developments and applications of the IEEE-1149.1 standard have enabled the use of JTAG in many other product life cycle phases. Boundary-scan technology is commonly applied to product design, prototype debugging, and field service as depicted in Figure 4.

The same test suite used to validate design testability can adapted and utilized for board bring-up, high-volume manufacturing test, troubleshooting and repairs, and even field service and reprogramming. The versatility of JTAG/boundary-scan tools delivers immense value to organizations beyond the production phase.

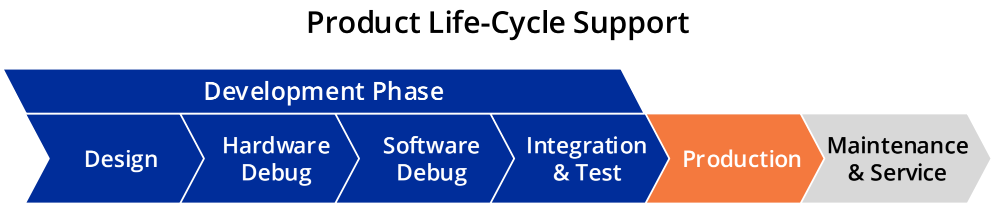


Figure 4. JTAG tools are used in all phases of the product life cycle.

JTAG Test Overview

Introduction

While originally developed to address the needs of testing printed circuit board assembly (PCBA) interconnects, JTAG test methods can be used to address many needs beyond simple structural test. This overview will briefly examine popular types of JTAG tests and applications.

JTAG Test Basics

Most JTAG/boundary-scan systems are composed of two main components: a test program generator for test development and creation, and a test program executive for running tests and reporting results.

Test Program Generator

Test program generators accept computer aided design (CAD) data as input in the form of a netlist, bill of materials, schematic, and layout information. The test program generator (TPG) uses the information provided in these files, along with guidance from the test developer, to automatically create test patterns for fault detection and isolation using JTAG-testable nets on the PCB. Full-featured test program generation software will generally also include the capability to automatically generate tests for non-scannable components including logic clusters and memories that are connected to boundary-scan devices. A sample of faults that can be detected with automatically generated tests is shown in Figure 5.

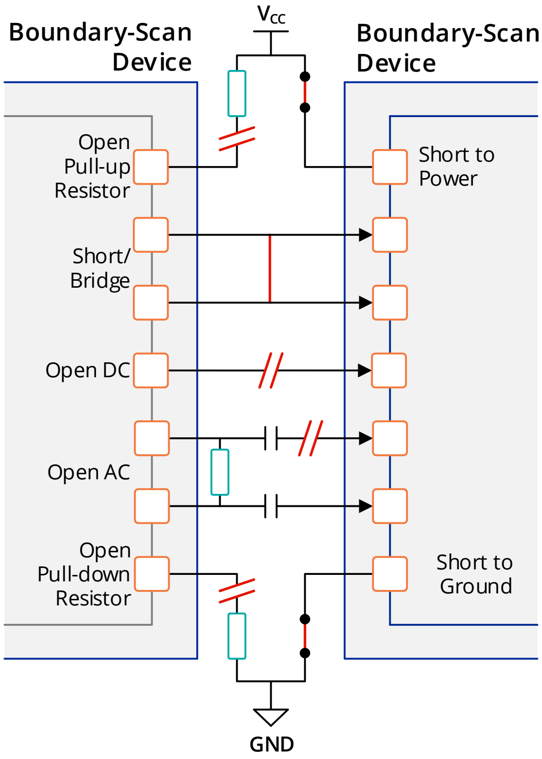


Figure 5. Sample of faults detected through JTAG test.

Test Program Executive

Test program executives are used to run the tests created by the test program generation software. The test executive interfaces with the JTAG hardware to execute test patterns on a unit under test (UUT), then compares the results with expected values and attempts to diagnose any failures. Modern test executives include advanced features such as flow control, support for third party test types, and often include an application programming interface (API) for integration with additional test systems or development of simplified operator interfaces.

JTAG Benefits

The continuous drive toward higher density interconnects and finer pitch ball-grid-array (BGA) components has fueled the need for test strategies that minimize the number of test points required. By embedding the test logic within the IC itself and limiting the physical interface to just a few signals, JTAG/boundary-scan presents an elegant solution to testing, debugging, and diagnosing modern electronic systems.

Today, JTAG provides the access mechanism for a variety of different system operations. Just some of the benefits provided by JTAG are:

Reuse through the product life cycle. The simple access mechanism provided by the JTAG TAP can be used at all stages of the product lifecycle—from benchtop prototype debugging to high volume manufacturing and even in the field.

Test point reduction. JTAG provides test access through just 4 pins (2 pins for IEEE-1149.7 compliant devices), reducing the number of test points required, resulting in lower PCB fabrication costs and reduced test fixture complexity.

Independent observation and control. Boundary-scan tests operate independently of the system logic, meaning they can be used to diagnose systems that may not operate functionally.

Extensibility. JTAG has seen continuous development and new applications are frequently being discovered. Additional standards have been developed to address AC-coupled testing, reduced pin counts, and control of test instruments embedded within ICs.

Scan Chain Infrastructure Test

JTAG testing usually begins by checking the underlying infrastructure to ensure that all devices are connected and test capabilities are operational. Test patterns are used to exercise the instruction register and boundary-scan register for comparison against expected lengths and values. If present, device ID codes can also be read and compared against expected values to ensure that the correct component has been placed.

Interconnect, Bus Wire, and Resistor Tests

After verifying that the scan chain is working properly, test patterns can be used to verify interconnectivity between system components. Nets that involve three or more boundary-scan pins represent a special case, called a bus wire, where additional patterns can be used to isolate faults to a specific pin, as shown in Figure 6. During a buswire test, boundary-scan driver pins are tested one at a time to ensure that all possible opens are tested.

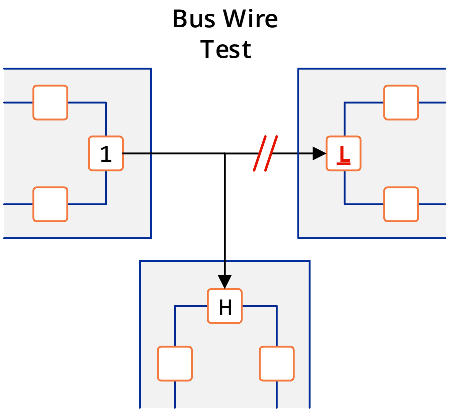


Figure 6. A buswire test can be used to diagnose open faults at the pin level.

Devices that are transparent to DC signals can be modeled as “short” signal paths and included in the test; for example, series resistors can be tested for component presence and open faults, while directional buffers can be constrained and tested to ensure that signals sampled at the buffer output pins match the signals that are applied to the buffer input pins. Additionally, tests for AC-coupled signals can be integrated with interconnect and buswire tests in systems with IEEE-1149.6 standard components, allowing capacitors to be tested for AC signal transparency.

Special tests can also be used to check pull-up and pull-down resistors, ensuring that resistors are present in the assembled system in addition to testing the nets for open and short faults. To accomplish this, resistors are tested by first driving the signal to a state opposite the pulled value. The net is then tri-stated, allowing the resistor to pull the signal back to the original state. Finally, the signal is sampled and the value is compared to the expected pulled value.

Logic, Memory, & Complex Devices

Not only can interconnections between boundary-scan components and simple transparent components be tested, but additional non-boundary-scan components can be controlled and tested for functionality and continuity using connected boundary-scan components. Simple test patterns may be used to test logic devices such as decoders or multiplexers, while sophisticated scripts may be used control and test complex devices for basic or advanced functionality, including analog-to-digital converters, UARTs, and Ethernet PHYs.

A common application of a cluster tests uses the storage capability of RAM devices to verify interconnects between a boundary-scan device and a connected memory, as shown in Figure 7. Using a model of the memory component, tests can be automatically created to write specific data patterns to memory addresses and then read back and compared against the expected value. These patterns are designed to ensure that all memory data and address signals are driven to both high and low logic states. The same concept used to test RAM can also be applied to non-volatile memory, such as flash, EEPROM, and NVRAM components.

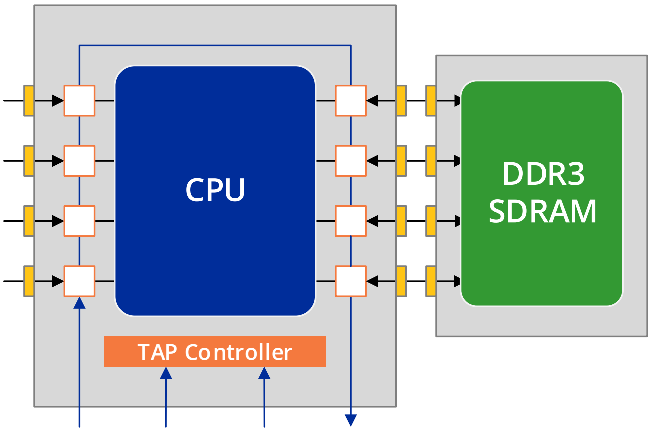


Figure 7. Memory interconnects are tested using a connected boundary-scan device.

JTAG Embedded Test

Many modern processors use JTAG as the main interface for on-chip debugging (OCD), allowing the processor to be controlled over the JTAG port within an embedded system. Using this same interface, the JTAG port can be used to initialize a processor, download and run a test program, and then obtain results; this test technique is a fast, convenient method for developing and executing peripheral tests and in-system-programming operations in embedded systems. Because these tests run at the system processor speed, defects that may not be identified during low-speed execution can be detected.

In-System-Programming

In addition to test applications, JTAG is also frequently used as the primary method to program devices such as flash memory and CPLDs. To program flash devices, the pins of a connected boundary-scan-compatible component can be used to control the memory and erase, program, and verify the component using the boundary-scan chain. FPGA and CPLD devices that support IEEE-1532 standard instructions can be accessed and programmed directly using the JTAG port.

Faster performance can be achieved using a CPU or FPGA to program the flash. In these cases, a small flash programming application is downloaded to the controlling device over the JTAG port, which is then used to interface between the test system and the flash programming application running on the embedded system. This configuration is depicted in Figure 8. This embedded JTAG programmer can run at much higher speeds than boundary-scan, increasing production throughput and rivaling or surpassing the speeds of USB and Ethernet-based programming solutions, without requiring an operating system or high-level software be present on the embedded system.

The IEEE-1149.1 JTAG team had the foresight to design an extensible standard—one that could employ additional data registers for many different applications. As a result, JTAG has grown from its original roots for board testing into a ubiquitous port that can be used for diverse applications such as in-system-programming, on-chip debugging, and more recently control of instruments embedded within ICs.

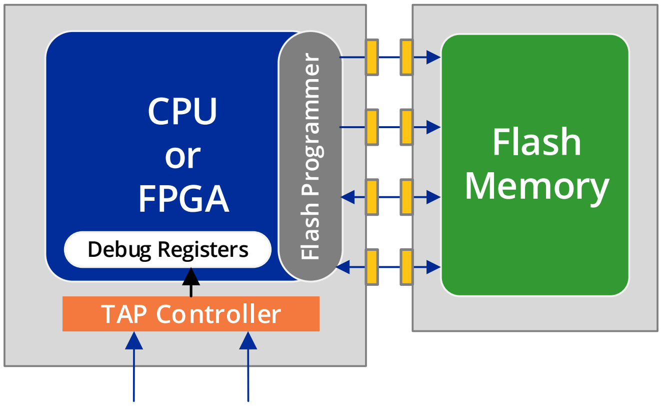


Figure 8. A CPU or FPGA under JTAG control can be used to program flash memory.

JTAG Technical Primer

Introduction

This primer provides a brief overview of JTAG devices--basic chip architecture, essential capabilities, and common system configurations.

JTAG Chip Architecture

The IEEE-1149.1 JTAG standard defines how IC scan logic must behave to achieve interoperability among components, systems, and test tools. ICs consist of logic cells, or boundary-scan cells, between the system logic and the signal pins or balls that connect the IC to the PCB. Each cell provides specific test capabilities—some cells can be used as input, others as output, and some are bidirectional.

The boundary-scan cells within a device are connected together to form a shift register, which is accessed through a serial test data input (TDI) and test data output (TDO) interface. The Test Access Port (TAP), consisting of 4 required signals and an optional reset signal, is the primary interface to the test controller which provides access to the logic.

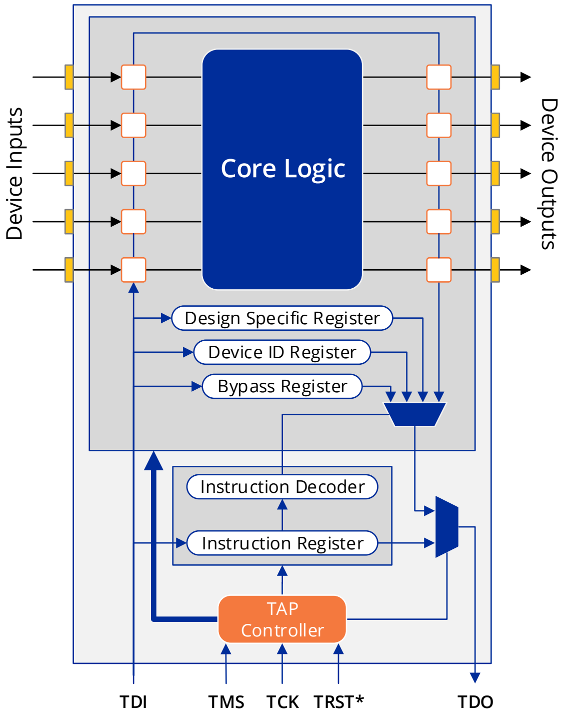


Figure 9. Diagram of basic JTAG IC architecture.

JTAG Instructions

IEEE-1149.1 specifies mandatory instructions—to be fully JTAG compliant, devices must utilize these instructions.

EXTEST

The EXTEST instruction is used to perform interconnect testing. When the EXTEST instruction is used, the mandatory boundary-scan register is connected between TDI and TDO and the device is placed in an "external" test mode. In this mode, boundary-scan output cells will drive test data onto the device pins and input cells will capture data from device pins—this is the main instruction used for boundary-scan testing.

SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is similar to EXTEST, but allows the boundary-scan device to remain in mission/functional mode while still connecting the boundary-scan register to TDI and TDO. When the SAMPLE/PRELOAD instruction is used, the boundary-scan register is accessible through data scans while the device remains functional. This is also useful for preloading data into the boundary-scan register without interrupting the device's functional behavior, prior to executing the EXTEST instruction.

BYPASS

When the BYPASS instruction is used, TDI and TDO are connected to a single-bit register that bypasses the longer boundary-scan register of the device—hence the name. BYPASS is very useful for reducing the overall length of a boundary-scan chain by eliminating devices that do not need to be involved in the current action. Devices that are given the BYPASS instruction remain in mission/functional mode while allowing serial data to flow through to the next device in the chain.

TAP Controller

The TAP controller as defined by the IEEE-1149.1 standard uses a 16-state finite state machine controlled by a test clock (TCK) and test mode select (TMS) signals. Transitions are determined by the state of TMS on the rising edge of TCK.

Two analogous paths through the state machine are used to capture and/or update data by scanning through the instruction register (IR) or through a data register (DR). The JTAG state machine is depicted in Figure 10 below.

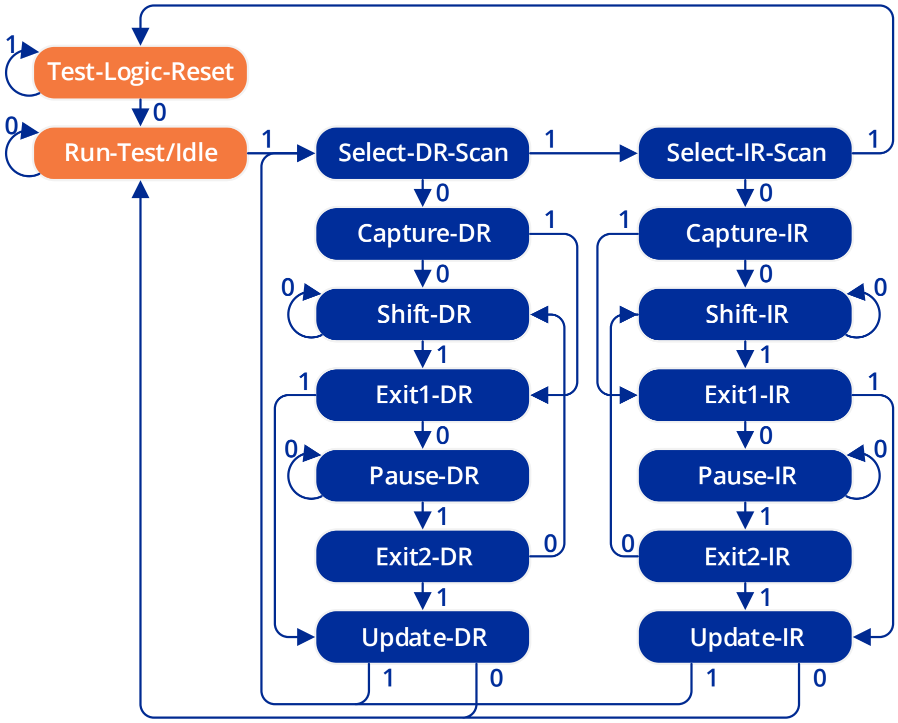


Figure 10. JTAG state machine diagram.

JTAG Interface

The physical JTAG interface, or test access port (TAP) consists of four mandatory signals and one optional asynchronous reset signal. Table 1 below summarizes the JTAG TAP signals.

Abbreviation	Signal	Description
TCK	Test Clock	Synchronizes the internal state machine operations.
TMS	Test Mode Select	Sampled at the rising edge of TCK to determine the next state.
TDI	Test Data In	Represents the data shifted into the device's test or programming logic. It is sampled at the rising edge of TCK when the internal state machine is in the correct state.
TDO	Test Data Out	Represents the data shifted out of the device's test or programming logic and is valid on the falling edge of TCK when the internal state machine is in the correct state.
TRST	Test Reset	An optional pin which, when available, can reset the TAP controller's state machine.

Table 1. TAP signal descriptions.

Many TAP interfaces will employ signals in addition to those required by the JTAG standard. For example, on-chip debugging applications may include signals for asynchronous halt and reset, while in-system-programming applications may increase programming speed by taking advantage of additional pins for time-critical functions such as toggling the write enable signal or polling a ready/busy signal.

JTAG Connectors

There is no single standard JTAG interface connector or JTAG pinout—physical characteristics such as pin spacing, interface voltage, and pin order vary among devices. Some TAP implementations may include additional signals such as a reference voltage, general purpose input/output (GPIO), or even serial bus signals, such as the example shown in Figure 11.

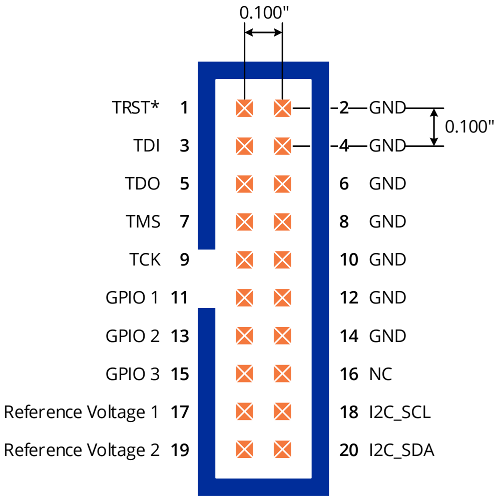


Figure 11. Typical JTAG connector diagram.

BSDL Files

Boundary-Scan Description Language (BSDL) files are used to describe the boundary-scan behavior and capabilities of a given device. Originally designed as a subset of VHDL, the BSDL format has been extended to add additional features and is not strictly VHDL compliant. The BSDL describes important properties of a given device's boundary-scan functions, including:

- Which JTAG standards are supported by the device.
- Signal mapping and package information.
- Available instructions, and which registers those instructions access.
- The type of boundary-scan cell available for each signal.
- Information about signals that affect compliance to the standard.
- Design warnings and notes.

Over time, the BSDL format has also been extended to include additional information; BSDL files may include descriptions of AC (IEEE-1149.6) testing capabilities, sequences in procedural description language (PDL) format, information about an electronic chip identifier (ECID), and more. BSDLs include statements that specify which standards are supported to allow automated tools to utilize supported features.

Scan Chain

JTAG devices may be daisy-chained within a system and controlled simultaneously. Boundary-scan test software can utilize one component to drive signals that will be sensed on a second component, verifying continuity from pin-to-pin. Devices can be placed in BYPASS mode to shorten the overall length of the chain to reduce test time. More complex designs may utilize additional circuitry or a dedicated JTAG bridge to selectively configure a scan chain that contains multiple devices, or even multiple sub-assemblies.

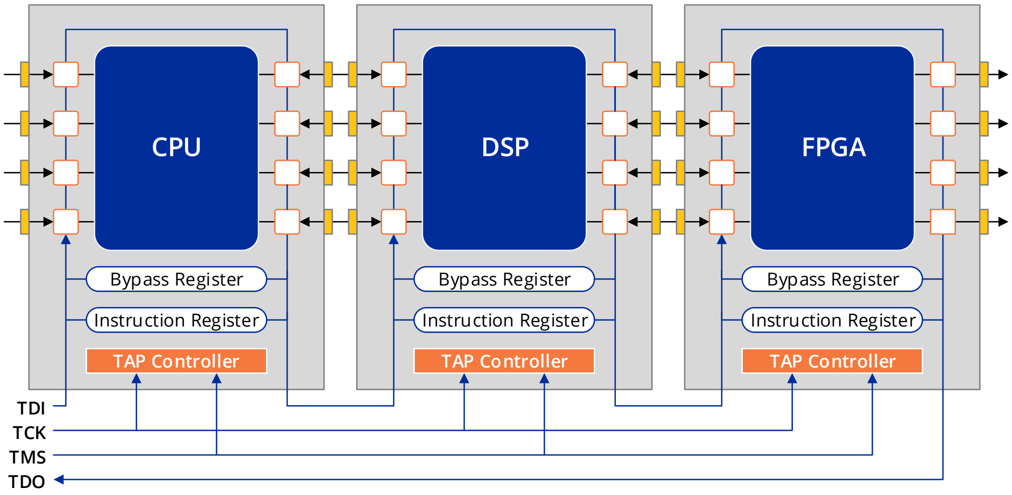


Figure 12. Example JTAG chain with multiple devices.

JTAG Applications

While it is obvious that JTAG based testing can be used in the production phase of a product, new developments and applications of the IEEE-1149.1 standard have enabled the use of JTAG in many other product life cycle phases. Specifically, JTAG technology is now applied to product design, prototype debugging and field service as depicted in Figure 13. This means the cost of the JTAG tools can be amortized over the entire product life cycle, not just the production phase.

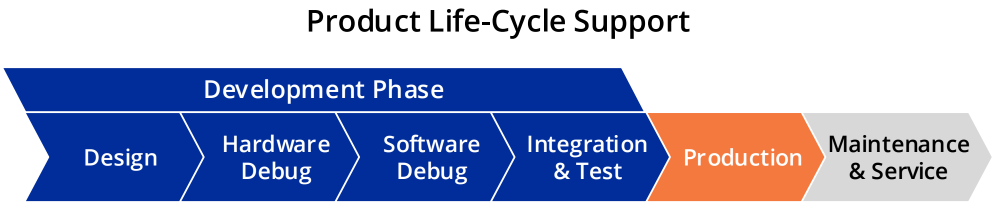


Figure 13. Product Life Cycle Support

To facilitate this product life cycle concept, JTAG tool vendors such as Corelis offer an integrated family of software and hardware solutions for all phases of a product's life-cycle. All of these products are compatible with each other, thus protecting the user's investment.

Applying JTAG for Product Development

The ongoing marketing drive for reduced product size, such as portable phones and digital cameras, higher functional integration, faster clock rates, and shorter product life-cycle with dramatically faster time-to- market has created new technology trends. These trends include increased device complexity, fine pitch components, such as surface-mount technology (SMT), systems-in-package (SIPs), multi-chip modules (MCMs), ball-grid arrays (BGAs), increased IC pin-count, and smaller PCB traces. These technology advances, in turn, create problems in PCB development:

- Many boards include components that are assembled on both sides of the board. Most of the through-holes and traces are buried and inaccessible.
- Loss of physical access to fine pitch components, such as SMTs and BGAs, makes it difficult to probe the pins and distinguish between manufacturing and design problems.
- Often a prototype board is hurriedly built by a small assembly shop with lower quality control as compared to a production house. A prototype generally will include more assembly defects than a production unit.
- When the prototype arrives, a test fixture for the ICT is not available and, therefore, manufacturing defects cannot be easily detected and isolated.
- Small-size products do not have test points, making it difficult or impossible to probe suspected nodes.
- Many Complex Programmable Logic Devices (CPLDs) and flash memory devices (in BGA packages) are not socketed and are soldered directly to the board.
- Every time a new processor or a different flash device is selected, the engineer has to learn from scratch how to program the flash memory.
- When a design includes CPLDs from different vendors, the engineer must use different in-circuit programmers to program the CPLDs.

JTAG technology is the only cost-effective solution that can deal with the above problems. In recent years, the number of devices that include JTAG has grown dramatically. Almost every new microprocessor that is being introduced includes JTAG circuitry for testing and in-circuit emulation. Most of the CPLD and field programmable array (FPGA) manufacturers, such as Altera, Lattice and Xilinx, to mention a few, have incorporated JTAG logic into their components, including additional circuitry that uses the JTAG four-wire interface to program their devices in-system.

As the acceptance of JTAG as the main technology for interconnect testing and in-system programming (ISP) has increased, the various JTAG test and ISP tools have matured as well. The increased number of JTAG components and mature JTAG tools, as well as other factors that will be described later, provide engineers with the following benefits:

- Easy to implement Design-For- Testability (DFT) rules. A list of basic DFT rules is provided later in this article.
- Design analysis prior to PCB layout to improve testability.
- Packaging problems are found prior to PCB layout.
- Little need for test points.
- No need for test fixtures.
- More control over the test process.
- Quick diagnosis (with high resolution) of interconnection problems without writing any functional test code.
- Program code in flash devices.
- Design configuration data placement into CPLDs.
- JTAG emulation and source-level debugging.

What JTAG Tools are needed?

In the previous section, we listed many of the benefits that a designer enjoys when incorporating boundary-scan in his product development. In this section we describe the tools and design data needed to develop JTAG test procedures and patterns for ISP, followed by a description of how to test and program a board. We use a typical board as an illustration for the various JTAG test functions needed. A block diagram of such a board is depicted in Figure 14.

A typical digital board with JTAG devices includes the following main components:

- Various JTAG components such as CPLDs, FPGAs, Processors, etc., chained together via the boundary-scan path.
- Non-JTAG components (clusters).
- Various types of memory devices.
- Flash Memory components.
- Transparent components such as series resistors or buffers.

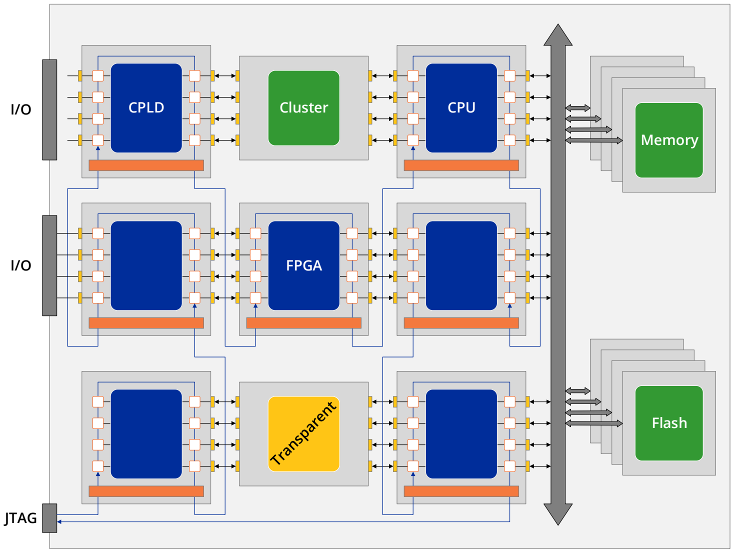


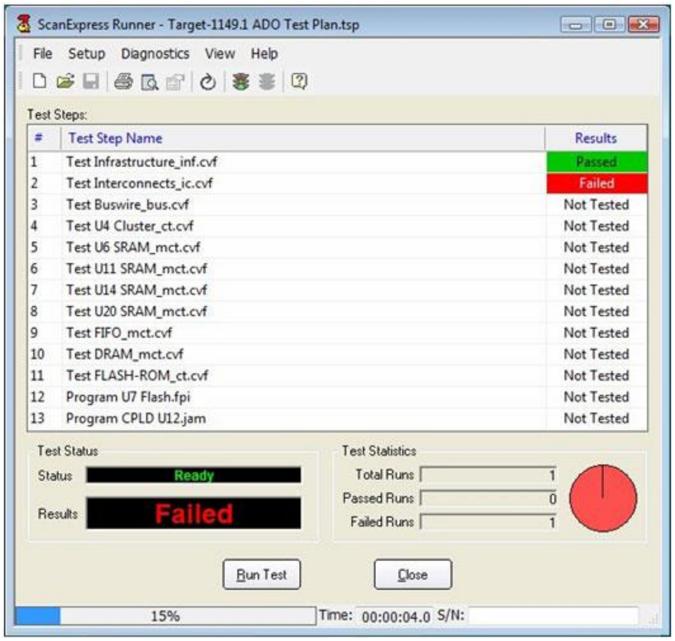
Figure 14. Typical Board with JTAG Components.

Most of the boundary-scan test systems are comprised of two basic elements: Test Program Generation and Test Execution. Generally, a Test Program Generator (TPG) requires the netlist of the Unit Under Test (UUT) and the BSDL files of the JTAG components. The TPG automatically generates test patterns that allow fault detection and isolation for all JTAG testable nets of the PCB. A good TPG can be used to create a thorough test pattern for a wide range of designs. For example, ScanExpress TPG typically achieves net coverage of more than 60%, even though the majority of the PCB designs are not optimized for boundary-scan testing. The TPG also creates test vectors to detect faults on the pins of non-scannable components, such as clusters and memories that are surrounded by scannable devices.

Some TPGs also generate a test coverage report that allows the user to focus on the non-testable nets and determine what additional means are needed to increase the test coverage.

Test programs are generated in seconds. For example, when Corelis ScanExpress TPG™ was used, it took a 3.0 GHz Pentium 4 PC 23 seconds to generate an interconnect test for a UUT with 5,638 nets (with 19,910 pins). This generation time includes netlist and all other input files processing as well as test pattern file generation.

Test execution tools from various vendors provide means for executing JTAG tests and performing in-system programming in a pre-planned specific order, called a test plan. Test vectors files, which have been generated using the TPG, are automatically applied to the UUT and the results are compared to the expected values. In case of a detected fault, the system diagnoses the fault and lists the failures as depicted in Figure 3. Figure 3 shows the main window of the Corelis test execution tool, ScanExpress Runner™. ScanExpress Runner gives the user an overview of all test steps and the results of executed tests. These results are displayed both for individual tests as well as for the total test runs executed. ScanExpress Runner provides the ability to add or delete various test steps from a test plan, or re-arrange the order of the test steps in a plan. Tests can also be enabled or disabled and the test execution can be stopped upon the failure of any particular test.



Test Steps:

#	Test Step Name	Results
1	Test Infrastructure_inf.cvf	Passed
2	Test Interconnects_ic.cvf	Failed
3	Test Buswire_bus.cvf	Not Tested
4	Test U4 Cluster_ct.cvf	Not Tested
5	Test U6 SRAM_mct.cvf	Not Tested
6	Test U11 SRAM_mct.cvf	Not Tested
7	Test U14 SRAM_mct.cvf	Not Tested
8	Test U20 SRAM_mct.cvf	Not Tested
9	Test FIFO_mct.cvf	Not Tested
10	Test DRAM_mct.cvf	Not Tested
11	Test FLASH-ROM_ct.cvf	Not Tested
12	Program U7 Flash.fpi	Not Tested
13	Program CPLD U12.jam	Not Tested

Test Status

Status: Ready

Results: Failed

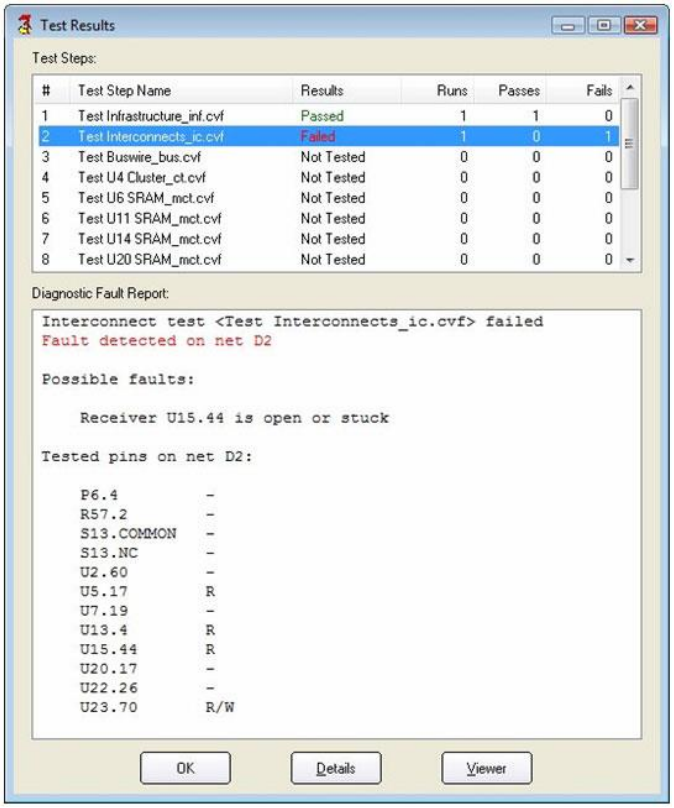
Test Statistics

Total Runs: 1

Passed Runs: 0

Failed Runs: 1

Figure 15. ScanExpress Runner Main Window.



Test Steps:

#	Test Step Name	Results	Runs	Passes	Fails
1	Test Infrastructure_inf.cvf	Passed	1	1	0
2	Test Interconnects_ic.cvf	Failed	1	0	1
3	Test Buswire_bus.cvf	Not Tested	0	0	0
4	Test U4 Cluster_ct.cvf	Not Tested	0	0	0
5	Test U6 SRAM_mct.cvf	Not Tested	0	0	0
6	Test U11 SRAM_mct.cvf	Not Tested	0	0	0
7	Test U14 SRAM_mct.cvf	Not Tested	0	0	0
8	Test U20 SRAM_mct.cvf	Not Tested	0	0	0

Diagnostic Fault Report:

```
Interconnect test <Test Interconnects_ic.cvf> failed
Fault detected on net D2

Possible faults:

    Receiver U15.44 is open or stuck

Tested pins on net D2:

    P6.4         -
    R57.2        -
    S13.COMMON   -
    S13.NC       -
    U2.60        -
    U5.17        R
    U7.19        -
    U13.4        R
    U15.44       R
    U20.17       -
    U22.26       -
    U23.70       R/W
```

Figure 16. ScanExpress Runner Diagnostics Window.

Different test plans may be constructed for different UUTs. Tests within a test plan may be re-ordered, enabled or disabled, and unlimited different tests can be combined into a test plan. ScanExpress Runner can be used to develop a test sequence or test plan from various independent sub-tests. These sub-tests can then be executed sequentially as many times as specified or continuously if desired. A sub-test can also program CPLDs and flash memories. For ISP, other formats, such as SVF, JAM, and STAPL, are also supported.

To test the board depicted in **Error! Reference source not found.**, the user must execute a test plan that consists of various test steps as shown in Figure 3.

The first and most important test is the scan chain infrastructure integrity test. The scan chain must work correctly prior to proceeding to other tests and ISP. Following a successful test of the scan chain, the user can proceed to testing all the interconnections between the JTAG components. If the interconnect test fails, ScanExpress Runner displays a diagnostic screen that identifies the type of failure (such as stuck-at, Bridge, Open) and lists the failing nets and pins as shown in Figure 4. Once the interconnect test passes, including the testing of transparent components, it makes sense to continue testing the clusters and the memory devices. At this stage, the system is ready for in-system programming, which typically takes more time as compared to testing.

During the design phase of a product, some JTAG vendors will provide design assistance in selecting JTAG-compliant components, work with the developers to ensure that the proper BSDL files are used, and provide advice in designing the product for testability.

Applying JTAG for Production Test

Production testing, utilizing traditional In-Circuit Testers that do not have JTAG features installed, experience similar problems that the product developer had and more:

- Loss of physical access to fine pitch components, such as SMTs and BGAs, reduces bed-of-nails ICT fault isolation.
- Development of test fixtures for ICTs becomes longer and more expensive.
- Development of test procedures for ICTs becomes longer and more expensive due to more complex ICs.
- Designers are forced to bring out a large number of test points, which is in direct conflict with the goal to miniaturize the design.
- In-system programming is inherently slow, inefficient, and expensive if done with an ICT.
- Assembling boards with BGAs is difficult and subject to numerous defects, such as solder smearing.

JTAG Embedded Functional Test

Recently, a test methodology has been developed which combines the ease-of-use and low cost of boundary-scan with the coverage and security of traditional functional testing. This new technique, called JTAG Emulation Test (JET), lets engineers automatically develop PCB functional test that can be run at full speed., If the PCB has an on-board processor with a JTAG port (common, even if the processor doesn't support boundary-scan), JET and boundary-scan tests can be executed as part of the same test plan to provide extended fault coverage to further complement or replace ICT testing.

Corelis ScanExpress JET™ provides JTAG embedded test for a wide range of processors. For more information about this technology and product, visit the ScanExpress JET product page.

Production Test Flow

Figure 5 shows different production flow configurations. The diagram shows two typical ways that JTAG is deployed:

- As a stand-alone application at a separate test station or test bench to test all the interconnects and perform ISP of on-board flash and other memories. JTAG embedded functional test (JET) may be integrated with boundary-scan.
- Integrated into the ICT system, where the JTAG control hardware is embedded in the ICT system and the boundary-scan (and possibly JET) software is a module called from the ICT software system.

In the first two cases, the test flow is sometimes augmented with a separate ICT stage after the JTAG-based testing is completed, although it is becoming more common for ICT to be skipped altogether or at least to be limited to analog or special purpose functional testing.

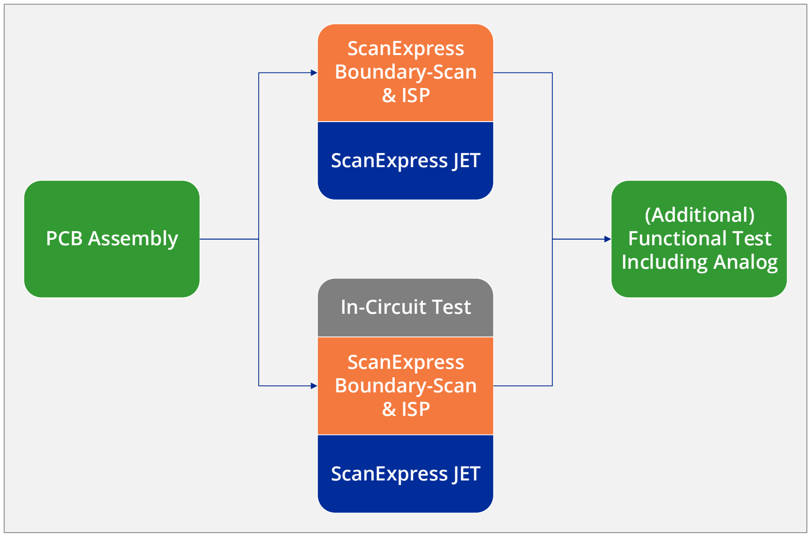

ScanExpress Boundary-Scan & ISP
ScanExpress JET
PCB Assembly
(Additional) Functional Test Including Analog
In-Circuit Test
ScanExpress Boundary-Scan & ISP
ScanExpress JET

Figure 17. Typical Production Flows.

The following are major benefits in using JTAG test and in-system programming in production:

- No need for test fixtures.
- Integrates product development, production test, and device programming in one tool/system.
- Engineering test and programming data is reused in Production.
- Fast test procedure development.
- Preproduction testing can start the next day when prototype is released to production.
- Dramatically reduces inventory management – no pre-programmed parts eliminates device handling and ESD damage.
- Eliminates or reduces ICT usage time – programming and screening.

Production test is an obvious area in which the use of boundary-scan yields tremendous returns. Automatic test program generation and fault diagnostics using JTAG software products and the lack of expensive fixturing requirements can make the entire test process very economical. For products that contain edge connectors and digital interfaces that are not visible from the boundary-scan chain, JTAG vendors offer a family of boundary-scan controllable I/Os that provide a low cost alternative to expensive digital pin electronics.

Field Service and Installation

The role of JTAG does not end when a product ships. Periodic software and hardware updates can be performed remotely using the boundary-scan chain as a non-intrusive access mechanism. This allows flash updates and reprogramming of programmable logic, for example. Service centers that normally would not want to invest in special equipment to support a product now have an option of using a standard PC or laptop for JTAG testing. A simple PC-based JTAG controller can be used for all of the above tasks and also double as a fault diagnostic system, using the same test vectors that were developed during the design and production phase. This concept can be taken one step further by allowing an embedded processor access to the boundary-scan chain. This allows diagnostics and fault isolation to be performed by the embedded processor. The same diagnostic routines can be run as part of a power-on self-test procedure.

JTAG Design-for-Test Basic Considerations

As mentioned earlier in this article, the design for JTAG test guidelines are simple to understand and follow compared to other traditional test requirements. It is important to remember that JTAG testing is most successful when the design and test engineering teams work together to ensure that testability is "designed in" from the start. The boundary-scan chain is the most critical part of JTAG implementations. When that is properly implemented, improved testability inevitably follows.

Below is a list of basic guidelines to observe when designing a JTAG-testable board:

- If there are programmable components in a chain, such as FPGAs, CPLDs, etc., group them together in the chain order and place the group at either end of the chain. It is recommended that you provide access to Test Data In (TDI) and Test Data Out (TDO) signals where the programmable group connects to the non-programmable devices.
- All parts in the boundary-scan chain should have 1149.1-compliant test access ports (TAPs).
- Use simple buffering for the Test Clock (TCK) and Test Mode Select (TMS) signals to simplify test considerations for the boundary-scan TAP. The TAP signals should be buffered to prevent clocking and drive problems.
- Group similar device families and have a single level converter interface between them, TCK, TMS, TDI, TDO, and system pins.
- TCK should be properly routed to prevent skew and noise problems.
- Use the standard JTAG connector on your board as depicted in Corelis documentation.
- Ensure that BSDL files are available for each JTAG component that is used on your board and that the files are validated.

Design for interconnect testing requires board-level system understanding to ensure higher test coverage and elimination of signal level conflicts.

- Determine which JTAG components are on the board. Change as many non-JTAG components to IEEE 1149.1-compliant devices as possible in order to maximize test coverage.
- Check non-JTAG devices on the board and design disabling methods for the outputs of those devices in order to prevent signal level conflicts. Connect the enable pins of the conflicting devices to JTAG controllable outputs. Corelis tools will keep the enable/disable outputs at a fixed disabling value during the entire test.
- Ensure that your memory devices are surrounded by JTAG components. This will allow you to use a test program generator, such as ScanExpress TPG, to test the interconnects of the memory devices.
- Check the access to the non-boundary-scan clusters. Make sure that the clusters are surrounded by JTAG components. By surrounding the non-boundary-scan clusters with JTAG devices, the clusters can then be tested using a JTAG test tool.
- If your design includes transparent components, such as series resistors or non-inverting buffers, your test coverage can be increased by testing through these components using ScanExpress TPG.
- Connect all I/Os to JTAG controllable devices. This will enable the use of JTAG, digital I/O module, such as the ScanIO-300LV, to test all your I/O pins, thus increasing test coverage.