Why 1149.1 (JTAG) Really Works

Gordon D Robinson

GenRad, Inc.
300 Baker Ave, Concord, MA 01742–2174

Abstract

The IEEE 1149.1 Boundary–Scan standard (JTAG) has now been a reality for over four years, and has had a lot of success, not always in the ways that the pioneers anticipated. This paper gives an overview of the important things that the standard does; the ways that this has proved to be useful; the importance of standardized descriptions for the components; and some of the reasons why using boundary–scan is not always as delightful an experience as users would like.

Introduction

Boundary Scan has been around for a long time. When the Joint Test Action Group (JTAG) started work in 1985 on what became the IEEE 1149.1 Standard Test Access Port and Boundary–Scan Architecture[1], the basic technology of boundary scan had already been used successfully by several different organizations. Despite its age and undoubtedly growing usage, many engineers in the electronics industry are still not familiar with what it is, why it is useful, and what has to be done to succeed with using boundary–scan.

Boundary–Scan Basics

The fundamental concept of boundary–scan is extremely simple. Test circuitry is placed just inside each pin of a component, that can either control or observe what is happening on the board or module that the component is being used in. There is also control logic that tells the test logic when to perform its test functions and when to let the component behave in its normal manner. The basic motivation for boundary–scan is the increasing density of components and leads that makes full bed–of–nails access difficult to provide.

The boundary–scan test logic means that test systems or subsystems can make things happen on the board, and see what is happening as a result without needing to know anything at all about what the logic inside the component is, or how it works. This capability makes the test methods that use boundary–scan completely independent of the complexity of the components in the unit being tested.

Boundary scan would have remained a niche technology without the IEEE standard. The developers of the Standard had several very clear goals:

- The Standard mandates enough test features in a component that incorporation of the Standard would always provide test benefits.
- The Test Access Port and its protocol are defined extremely precisely such that interoperability is assured for configurations of compliant components, and for the test equipment accessing the configuration.
- The definition of the boundary–scan instructions is robust for use in many test applications.
- The Standard does not limit the test features that a component can provide; but does define the mechanism for providing additional features. The Standard also defines optional facilities that are very useful.

An oversimplified view of those goals was that, if components do everything that the standard says they have to, then those components can be used for test purposes without any problems.

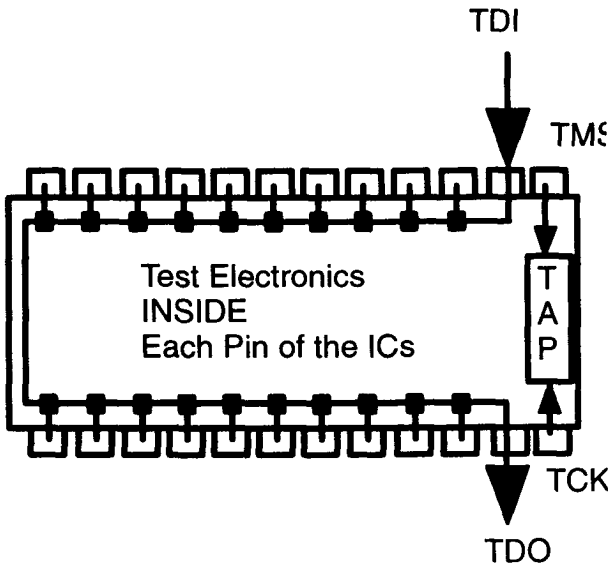

Fig 1: Boundary–Scan Architecture

There are many ways that those goals could be achieved: the standardization process chooses one of those ways. Not everyone agrees with the tradeoffs that were made, but the advantages of working with a well–supported Standard normally justify the costs and difficulties of adhering to the Standard.

What Makes JTAG Work?

JTAG works because of several complementary factors.

- The interoperability that adhering to the Standard provides.
- The fact that boundary–scan gives value in many different test environments.
- The availability of a language (BSDL) that describes the boundary–scan features of a component to test software.
- The availability of BSDL descriptions for many components.
- Tools and test systems that make effective use of the added test facilities.

Overview of 1149.1

The IEEE Standard is well over 100 pages long, and is very detailed in many places. It is fairly complex, despite the developers' desire to make the Standard easy to understand (so that it would not be misunderstood). Any summary fails to describe important features of the full definition. This description deliberately omits a lot of detail to describe the basic concepts of the Standard in a tutorial manner (see Fig 1).

Test Access Port and Protocol

The Test Access Port (TAP) is the set of signals used to control the test features and get test results from the devices. The TAP signals of components normally form a simple chain with the same Mode (TMS) and Clock (TCK) to all of the components, and with Test Data Out (TDO) of one component going to Test Data In (TDI) of the next. The protocol provides four facilities.

- Reset the test logic. The component behaves normally.
- Load new instructions into each component. Each component can be given a different test instruction.

- Data Scan operation: this consists of three stages: the capture of information on the register inputs; shifting out of the captured data while shifting in new data; and the update of the register outputs with the newly shifted data.
- Run Test: stay in a state and run any special test features on each test clock. This facility is also often called Idle, because that is all that happens for most of the standardized instructions.

Mandatory Instructions

All 1149.1 components must implement three instructions: EXTEST, SAMPLE/PRELOAD and BYPASS.

The EXTEST instruction makes the component do its fundamental boundary–scan test function: the cells on input pins act as sensors, while the cells on output pins act as drivers.

The SAMPLE/PRELOAD instruction is mainly used to preload the boundary register with a known value, so that there is a defined effect when the EXTEST instruction is loaded.

The BYPASS instruction makes the shift path through the device into a single bit register, the BYPASS register, but otherwise leaves the device in its normal operating mode.

Optional Instructions

Any 1149.1 component can implement any additional instructions. The standard describes several that are frequently implemented.

The HIGHZ instruction puts all of the device outputs into a tristate mode; which is very useful for in–circuit testers to avoid stress as the tester tests adjacent components.

The CLAMP instruction sets all of the device outputs into a fixed state, just like the EXTEST instruction.

Both HIGHZ and CLAMP put the BYPASS register into the shift path, so no data needs to be shifted to preserve the static state.

The RUNBIST instruction causes the component's self–test procedure to be run. This is the most common way of testing the internal logic in 1149.1 devices: self–test is fast and uses only a small amount of test data.

The IDCODE and USERCODE instructions read a value from a device identification register that identified the type of device (IDCODE) and the program in the device (USERCODE) for programmable logic. These are extremely useful instructions for identifying that the wrong component has been placed on a board.

The INTEST instruction acts like the opposite of EXTEST. The boundary register is used to test the system logic. This instruction is rarely implemented.

The Boundary Register

The boundary register in an 1149.1 device contains the cells that act as drivers, sensors and drive enable signals: these latter cells are called control cells.

Most boundary register cells consist of two parts, the shift cell, which is the path for data capture and shifts, and the latched parallel output which only gets new values during the update step of a Data Scan.

The latched parallel output, along with the fact that the whole boundary register must be dedicated test logic, is one of the more controversial features of 1149.1. Its purpose is to ensure that the boundary scan tests never suffer from the device signals "rippling" during data shifts, causing very unpredictable results on the logic being tested through the boundary–scan logic.

The Variety of JTAG Test Applications

One of the most important reasons for the growing use of JTAG and its acceptance by the industry is that it provides substantial benefits in many different test environments[2].

- A single Boundary–Scan part on a board allows an in–circuit test for that component to be produced very rapidly. The test will be easier to develop, run more quickly, and give more accurate diagnosis than a conventional in–circuit test.
- When many JTAG parts are interconnected, the Boundary–Scan capabilities can be used to test for opens and shorts on a board test system. By using the tester access to internal nodes, the test can ensure that there is negligible loss of fault coverage.

- When many JTAG parts are interconnected, the Boundary–Scan capabilities can be used to test for opens and shorts on the boundary–scan nodes from the edge of the board, with reasonable fault coverage, that depends on the proportion of boundary scan nodes.

- The self–test facilities in components can be activated. This leads to the basic strategy of using boundary–scan for interconnect test, plus component self–test, as the modular self–test strategy at any integration stage.

- When a non–boundary–scan cluster is surrounded by boundary–scan parts, it is often possible to use the boundary–scan parts to apply a static (slow speed) test to the cluster and see the results.

- Some programmable devices can be reprogrammed through their JTAG ports.

- On–Line system maintenance, using the boundary–scan facilities to test a part of the system without removing it from the system.

Descriptions of Boundary–Scan Parts

Boundary–Scan testing is performed indirectly: there is a serial path on the UUT that provides access to many parts of that UUT. The test tools that use that path depend on accurate descriptions of the boundary–scan facilities.

At around the time that the Standard was first approved, several of the companies that were planning to develop tools that supported the Standard realized that it would be in everyone's best interests if there were a single way to describe these parts to all of the software tools. The BSDL language has come from those efforts. BSDL is currently being standardized by the 1149.1 working group in the IEEE, and has been a de–facto standard since 1991.

Because the whole tool industry uses a single language for describing the boundary–scan parts, the task of producing BSDL for a new component is seen as the responsibility of the semiconductor companies that design the parts. Many of the semiconductor companies make BSDL available, either on disk, or from bulletin boards or Internet ftp sites. This availability has helped reduce the number of problems caused by incorrect BSDL descriptions for the components.

Obstacles Delaying Boundary Scan

Although the use of boundary–scan is increasing, there are several significant obstacles that are slowing down its progress.

Accidental Non–Compliance

The 1149.1 Standard is not a simple document to read, and it is definitely difficult to implement accurately in a component. As a result, many of the early 1149.1 devices had design errors such that the component was close to implementing the Standard, but missed on some details.

Unfortunately, a few details being wrong has a major impact on interoperability. And one of the pieces of interoperability is the test equipment and tools that expect the devices to adhere to the Standard.

One of the worst forms of accidental non–compliance is when the basic TAP operations are not performed correctly. These design errors can make the test facilities in a device extremely difficult, and make it impossible to use the device on a chain with other boundary–scan parts.

Fortunately, the simplest form of test using boundary–scan, testing an isolated component on a board with full tester access, is extremely robust when such mistakes have been made[3]. The in–circuit test model is usually modified quite easily to avoid the areas where the design is in error. There is often a small loss of fault coverage, and always additional test development time, but most of the benefits are still obtained.

Deliberate Non–Compliance

Deliberate failure to comply with the Standard, while making claims of "compatibility" or "compliance" is, in the author's opinion, the greatest obstacle to widespread adoption of boundary–scan test methods.

There are several attitudes that designers or their companies have taken to such non–compliance.

- Arrogance: shown by such statements as "We know this is not compliant, but we've done as much as we intend to: it's someone else's problem".

- Denial: "We'd like to know how BSDL and your tools can handle the *special features* of our implementation".

- Apology: "We wanted to comply, but some of the requirements conflict with the most important product requirements, so we have had to violate the Standard like this."

Few of those who state that the Standard is excessively burdensome have ever come to a working group meeting to make a case for the Standard becoming easier to implement.

BSDL Does Not Support Non–Compliance

One problem that all the non–compliant devices have, shared by the companies that design such devices as well as the companies trying to use the devices, is that the BSDL language does not allow non–compliance to be described.

This is a deliberate decision by the 1149.1 working group, based on the fact that if BSDL permitted the description of some forms of non–compliance, that would imply to the industry that those parts of the Standard were optional, not mandatory.

Common Non–Compliance

Several forms of non–compliance occur very frequently, particularly in deliberately non–compliant devices.

- Pins without scan access. The Standard mandates that all digital pins except for the TAP must have boundary–scan cells. Many devices have a small number of pins that do not have such access. This is a mild non–compliance, as test systems can pretend that such pins are not digital pins.

- No update latches in the boundary register, or a more severe case of that, a boundary register that is not dedicated test logic. This causes the rippling as data values are shifted that the Standard regards as unacceptable. Many designers believe that this is unimportant, because they know what the devices will be connected to, and are sure that the rippling will not cause problems.

- Clock requirements. The Standard is very strict about there being no synchronization needed between the test clock and the system clock for a component; but some devices impose such constraints for the boundary–scan logic to work. Some devices merely require a fast enough free–running clock; others have specified the way that the test clock must be synchronized.

All non–compliance carries risk: every developer of boundary–scan tools is making some assumptions about what forms of non–compliance their tools will tolerate, but these decisions are not often made public, and without any guarantees of support of the non–compliance with future features.

Proving the Value – Price of JTAG

JTAG is not free. Sometimes it appears that it is, because a device has JTAG incorporated, and there is no equivalent device available. This is how many organizations start to use it: it is there and gets used by the test systems without their trying.

When there is a choice between a JTAG device and a non–JTAG device for the same mission function, the additional cost of the JTAG components sometimes prevents designers from using them. The extra test development costs are smaller than the additional component costs for each board made. This economic argument depends on the number of test stages visible to the designer making the choice.

Sometimes there is an alternative to JTAG, that costs less and works well enough in the anticipated test environments. The best known of these is the NAND tree, that is very useful for detecting many simple faults in full access board test. As these trees are often present in ASIC devices, because the vendor insists on their being implemented, designers need a lot of convincing to add JTAG in addition to the NAND tree.

There is one other price attached to JTAG, particularly for ASICs. That is time to market: the additional design effort for JTAG becomes the critical path for the chip design, and it is the first feature dropped to get the project back onto schedule. The recent introduction of tools that will synthesize the JTAG circuitry (and also produce the BSDL) should eliminate this problem, as well as

decreasing the number accidental or deliberate violations of the Standard.

Where Next for 1149.1?

The 1149.1 Working group has been working on the Standard since 1988. Before that the JTAG organization had spent three years developing the base specification. The 1149.1a–1993 supplement to the Standard was long, mainly because the most complex chapter in the Standard was rewritten for clarity; and the BSDL supplement (1149.1b) is a substantial document (over 60 pages). The BSDL supplement is expected to become an IEEE Standard during 1994.

A likely path for the Working Group to take then is to leave the Standard alone until there is a good reason to change it.

One possible good reason would be that another test bus Standard wants to use 1149.1 as its base, and so either the new test bus work gets merged into 1149.1 or the 1149.1 Standard is restructured so that the common part between the Standards is clearly defined.

Summary

Today, 1149.1 Boundary–Scan (JTAG) is reaching the end of its major development phase, and becoming a mainstream technology that can be used readily. It works because there is a Standard that gives interoperable access to useful facilities, and a method, BSDL, for getting the description of the variable parts of those facilities into available test tools. Production of accurate BSDL is seen as a task for the component designer, and the semiconductor industry and ASIC design tools have responded by producing BSDL. There are still not enough convincing arguments to show that 1149.1 boundary–scan always saves money over the product lifetimes; that argument depends how many test environments the boundary–scan is used in, and is made harder by the companies that deliberately violate 1149.1, while claiming to adhere to the Standard.

References

[1] IEEE Std 1149.1 – Standard Test Access Port and Boundary–Scan Architecture.

[2] GenRad, Inc. Meeting the Challenge of Boundary–Scan.

[3] Albee, A.J. et al; "Basic Boundary Scan for In–Circuit Test"; Proceedings 1993 European Test Conference.