print | close

# IEEE Std. 1149.1: This Isn't Your Father's JTAG Anymore (Part 1)

*Electronic Design*

Carl Barnhart, Bill Bruce, Alan Bair, and Jim Johnson, SiliconAid

Fri, 2014-11-14 12:01

Literally, IEEE Std. 1149.1 very well might have been your father's JTAG, since it has been around since 1990. For over 20 years, this standard has been in use throughout the world (boundary-scan description language, or BSDL, wasn't added until 1993). It was revised in 2001 with only incremental improvements and no real major rewrites—until now. With the new 1149.1-2013 and the upcoming release of IEEE Std. P1687, huge changes and unmistakable momentum are on the horizon, promising to once again revolutionize the industry.

This is the first article in a series aimed at discovering a new approach to an old problem: "How to leverage reuse and automation on an industry scale to both save and make more money?" The series will focus on the newly revised IEEE Std. 1149.1 standard, as well as the proposed IEEE Std. P1687 standard.

The new IEEE Std. 1149.1-2013 standard has more than doubled in size from the previous release. It includes many new features, such as automated component initialization, complex and variable-length user test data register structures with excludable (included or excluded) and selectable (one of n segments selected) register segments, new structural documentation, and a procedural language for describing the use of the test structures. In addition, expanded documentation capabilities allow an intellectual-property (IP) supplier to hierarchically document both the test structures built into the IP and the procedures for using those test structures.

The IEEE working group's P1687, under development for several years, is dedicated to defining access structures (control and observe) and documentation for internal chip "instruments." It defines the structure and documentation of a potentially complex network of instruments whose access via the network can be reconfigured. Currently, the P1687 uses the 1149.1 TAP for access to the network, appearing to the 1149.1 TAP as a variable-length user test data register. Both standards allow and support a hierarchical description of networks, enabling the chip integrator to utilize a hierarchical network to connect IP blocks supplied by IP providers.

Later in the series, we'll discuss how these two new standards will be used, who needs them, and how the industry in general may be changed in the coming years by their mutual existence.

## A Little History Behind IEEE 1149.1-2013

When the revision that became known as IEEE Std 1149.1-2013 got underway, the only "big" issue was a long-standing need to provide some form of automated initialization for the newer complex I/O. Through investigation, it became clear that two things first had to be standardized: a way to document the multiple fields of a test data register (TDR); and a way to document the procedures needed to initialize a component on a board.

The ability to document the multiple fields of a TDR has been around forever, but not in machine-readable form. Anyone with TDR programming experience has used specification documents to find the parameters of

isters, such as how the multiple fields are laid out, what each field does, and what legal values can be used for h field.

It's now possible to document all of those parameters in the machine-readable BSDL description of a component. By assigning names for the fields, names (mnemonics) for the legal values, and associations of fields to an I/O, one can logically organize these field descriptions by grouping them into TDR segments. In addition, one can now assemble previously described TDR segments, including segments defined by an IP provider, into larger segments or entire TDRs.

As for documenting the procedures needed to initialize a component on a board, the 1149.1-2013 working group adopted an initial IEEE Procedural Description Language (PDL) that's under development as a starting point. This version of PDL was then modified to meet the needs of the 1149.1-2013 group. Due to the different requirements of 1149.1-2013 and P1687, each group developed their own flavor of PDL. Unfortunately, due to the many differences between the resulting PDLs in syntax and usage, they will need to be treated as two different languages. Only a subset of syntax can be shared between both PDL languages.

PDL is designed to document the procedures for stimulating and observing test data register fields for 1149.1-2013. In the case of P1687, the PDL documents the procedures for stimulating and observing data to an instrument. This is a subtle, but important difference between how the two standards define the PDL interface. Both PDL languages are very flexible and can support the names of the register fields, including the names of the values (mnemonics) without regard to which bits of which register were being written or read. With restrictions, the PDL commands can be combined with Tcl to further improve control and flexibility when stimulating and observing the registers and instruments.

At this stage, a number of questions started popping up, such as:

- What about chips with multiple power domains, domains that might be de-powered during test? To support that, the standard needed segments of TDRs (including the boundary-scan register) that could or couldn't be excluded.
- What about IEEE Std 1500 wrapper structures embedded in a TDR? To support that, the standard needed the ability to select one of many segments for inclusion in a TDR.
- What about all of the IP being used, much of it containing I/O and/or TDR segments? To support IP, the IP supplier should be able to provide the documentation. Thus, the BSDL Package and PDL were modified to allow hierarchical description of both the embedded TDR segments and necessary procedures.

All of these capabilities were generalized to make them useful in as many situations as was practical. By this time, the working group had developed documentation standards for complex TDRs and procedures, resulting in a very robust initialization capability!

In the meantime, the P1687 working group, often known as iJTAG (Internal JTAG), was developing its own standard. It focuses on documenting the structural and procedural descriptions used to access a possibly complex network of internal "instruments" on the component.

Related

What's The Difference Between JTAG (IEEE 1149.1) And IJTAG (IEEE P1687)?

IJTAG Interoperability Fosters Two-Way Flow for Debug, Verification

What's The Difference Between Scan ATPG And IJTAG Pattern Retargeting?

network to access the register fields is described with BSDL extensions for 1149.1-2013 and a new language ed Instrument Control Language (ICL) for P1687. While both of these can describe very complex networks, ICL modeling language can describe some networks that are beyond the capability of 1149.1-2013. These differences will be discussed in a future article.

As a result, considerable overlap exists between the capabilities of the two standards when documenting user-defined TDRs and procedures. In a sense, a continuity of capability prevails from the simpler configuration to a more complex configuration. Test features usable for component or board test, in particular, may be simply added to the otherwise required 1149.1 BSDL. A more complex configuration with an extensive set of instruments, many intended for system performance monitoring rather than manufacturing tests, may be supported by either standard. Finally, P1687 supports some network configurations not supported in 1149.1-2013. These tradeoffs will also be worked through in detail later in the series.

## Buzz-Worthy P1687

For several years, IEEE Std. P1687 has created a lot of buzz around our industry. A great deal of work, over many years of development, has companies anticipating its release as a standard. That said, many companies have already designed and implemented 1687-like structures into silicon; even before the standard has been published. EDA companies, such as SiliconAid, have invested years of development in tools to support the standard. Some tools have actually been released to the market and are already available for purchase.

P1687 started with a simple idea: Define the interface and access to a piece of internal IP. Then things started to evolve. The development of enhancements, add-ons, feature-creep, and more, plus throw in some unique personalities and debates, and now we've reached this point. We're still waiting (and hoping) to get P1687 released sooner than later. Obviously, though, a lot of technical issues still must be considered in the development of this standard.

The first version of P1687 supports the IEEE 1149.1 Test Access Port (TAP) standard test interface. Future possible versions also may support other popular interfaces.

## Impact of These New Standards

Let's table the debate on comparing these standards a little while longer. Instead, we'll discuss how these two standards might affect things in the future. Reuse, automation, time to market, cost savings, higher quality, and efficiency are all terms usually used by sales and marketing types to try to entice you into buying their products. However, these terms can also be used to describe some of the real benefits gained by adopting one or both of these standards.

In the design phase of a chip, IP is typically developed internally or it comes from third-party providers. The IP is then integrated into the architecture to functionally perform the task needed by the end application. These can be complicated IP like DDR, SerDes, PLLs, memories, CPUs, etc. They can also be simple IP, such as voltage monitors or temperature monitors to check the fab process. You can imagine dozens, or even hundreds, of pieces of IP sprinkled throughout a large customer chip or SoC.

A huge task for any chip-verification team is to develop all of the tests and the testbench to verify all of a SoC's IP. Typically, they would have to figure out a way to simulate these IP and then observe something to indicate if it works or not. This could be some type of functional test, built-in self-test (BIST) or loopback test, CPU-based test, etc. They'll require knowledge of that specific chip architecture and how each IP should work. In many cases, these verification tests may need to be driven and observed from the chip pins in simulation; therefore, manufacturing tests can be developed from the resulting evcd file(s).

tainly, scan- and BIST-based testing has helped significantly to ensure that the manufactured digital logic is h quality. However, there's still the big issue of how to connect to these structures, control them, and get failing data out, let alone how to control the start and stop of these tests.

For IP, that might be timing sensitive or mixed signal. The simulation and ATE tester worlds may still need other types of verification and testing, though.

Wouldn't it be nice to reuse the original IP provider's original verification and test suites to make sure all of your IP works and was manufactured correctly? Correct for simulation, as well as for ATE test, board test, system test, and possibly in field testing. In this scenario, you're talking about an entire ecosystem, from design to application.

Wouldn't it also be nice to have a network that could be reconfigurable to get to and from all of these different pieces of IP in any order you like? This would allow you, for instance, to optimize which manufacturing tests are run, and in which order, based on test time, power, etc. All while minimizing test time by not having to access every IP on every scan frame.

Furthermore, wouldn't it be nice to not have to develop a testbench with all of the simulation and have it auto-generated for you?

Still better, wouldn't it be nice to get ATE patterns directly out of the same process without having to use additional tools to convert simulation to ATE patterns?

Finally, wouldn't it be nice to be able to use this same common technique and methodology across different products within your company? A common test methodology would give users the ability to standardize on a common test interface and common reusable test controllers to optimize flow, equipment, training, etc.

These issues and more can be addressed by using these two new standards.

## The Bloody Crusades of Embedded IP TEST

Because of the overlap between the new 1149.1-2013 and the P1687 standards, the obvious question is, "Which one is superior?" Though a better question could be, "Which standard better fits my goals and architectural requirements with the time and resources we have?"

IEEE 1149.1 is still the standard for traditional board test. Your board-mountable components will still rely upon it. However, the overlap between the two standards has a lot of people confused. If you take a step back, it becomes obvious that each can have a place in our new plug-and-play, automated, reusable world of embedded test capabilities. Both can also leverage integration, verification, ATE test, and validation within chips, boards, and systems.

Each standard has a few common basic ideas:

- Support of a reconfigurable network to access test capabilities
- Reuse of IP level patterns at chip, board, and system levels
- Standard JTAG test interface

It's important to remember that each standard has its own strengths, as well as slightly different input requirements. In future articles, we'll discuss how you might decide on the one that best fits your needs for a specific device and its goals.

ntually, the industry will sort out which standard is most useful in which situations. In the meantime, ugh, a bit of a "religious" debate already is building around this question. Many IP and components may profit from supporting both standards until the debate is settled. The information to do so already exists, and in most cases, simply needs to be formatted in the languages specified by the two standards. We may indeed find that in the designs of tomorrow with a mix of IP from multiple sources—some are 1149.1-2013-based, some are P1687-based, and some will support either.

## Conclusions

It is clear that both of these standards will affect you in the coming year or so if you are a part of the semiconductor industry. This is especially true of those of you in IP development, chip design, verification, integration, test, validation, board test, applications, or system development. Don't wait around to get involved. Now is the time to get on board before this train leaves you behind!

Watch for our next article in the series, coming soon.

*SiliconAid Solutions Inc., Austin, Texas, has contributing members in the following IEEE working group: 1149.1-2013, P1687, and IEEE 1149.6. SiliconAid provides a full suite of chip focused products to support P1687, 1149.1, and 1149.6 standards. (i.e., we are not a board test focused company, but rather a JTAG-standards-based, chip-focused company.)*

*SiliconAid Solutions is a provider of Design for Test consulting services encompassing methods, implementation, and team augmentation and training. The Senior DFT Consulting Group has been recognized for comprehensive support of all standard EDA DFT solutions. The company also delivers chip-focused JTAG software solutions to validate, verify, and utilize IEEE 1149.X- and P1687-related industry standards.*

**Source URL:** http://electronicdesign.com/eda/ieee-std-11491-isn-t-your-father-s-jtag-anymore-part-1