Article 3 – 1687 vs. 1149.1-2013 for Embedded IP

Authors: Carl Barnhart, Jim Johnson, Alan Bair, and Bill Bruce

Comparing and contrasting these two overlapping standards seems to have become almost a religious debate within our industry for the last few years. There are probably a few reasons why this is a polarizing topic.

The standards working groups are made up of people. Working on a standards working group is not a glory job or usually not even a fun job. Sometimes, it is a thankless job with a lot of hard work and dedication required over several years before a new standard can be published.

At the end of the day, people are human. People have personalities, personal interests, professional goals, a lot of pride in the work they have done, a wish to truly impact the industry in a positive way, and sometimes a vested business interest to make the standard a success.

Thanks again to all the standards working group members and the people behind the scenes for all your hard work and dedication! Hopefully both of these standards will move the industry forward.

Overview

This is the third article in a series of articles on a new approach to an old problem: "How to leverage reuse and automation on an industry scale to save/make more money?" This series of articles will focus on a newly revised IEEE Std. 1149.1-2013 and IEEE Std. 1687.

Hopefully our first two articles have given you a sense of how these two new standards might be used, especially for testing of internal IP. We also stated in a few places some of the basic differences between the standards. In this article, we will try to highlight most of the major differences and how they are relevant in different situations. We'll discuss these situations that may occur and which standard might have advantages for your specific requirements.

Please note that 1149.1-2013 addresses many other new features in addition to the sections applicable for internal IP. 1687 is dedicated to internal IP test structures and methods.

Authors Perspective

This series of articles is meant to be very objective. We want to take out the religious debate and be the "neutral Switzerland" in the discussion. We have been highly involved in both standards for many years. We will not focus on how the standards evolved and instead focus only on what they are today and the best ways to utilize them.

SiliconAid Solutions has a vested interest in using and promoting both IEEE Std. 1149.1-2013 and the IEEE Std. 1687. Since we have been involved with experts over many years, we believe we can represent both bodies of work well.

Getting to Switzerland

We quickly found writing this series of articles that each standard can be interpreted slightly differently. Trying to compare and contrast two standards of hundreds of pages each is not an easy task. We also wanted to make sure that our interpretations were correct and that others involved in the working groups agreed. So we started dialogs with some other members of each working group on specific points where the two standards differed. We found that even members of the same working group didn't agree on all aspects in the interpretations of the standard and how to use it. **Special thanks to all the people that talked with us to improve the content of this article!** We found that only through detailed discussions on how and why each standard could be used and in which situations did we start to understand each point of view.

In this article, we will do our best to represent the facts that are written in the standards, offer a few suggestions, and leave most of the interpretation to you. Certainly for a lot of points, you may find advocates of one of the two standards that come up with creative solutions to solve issues. We'll try and explain the tradeoffs without a lot of extreme modifications to "fit" it into a standard. An example would be you that might be able to use this standard if you always did some type of elaborate sequence or modified your flow to do this. **In keeping with our belief in simplicity, these types of major work-arounds will not be discussed in this article.**

In this series we'll discuss how these two new standards may be used, who needs them, and how the industry in general may be changed in the coming years because of these standards.

Deciding which standard can work for you

We estimate that either 1687 or 1149.1-2013 can work for the majority (>80%) of designs or SOCs with regards to embedded IP. Remember, the instrumentation functions of 1149.1 are a subset of 1687 with some exceptions such as applying data constraints that we'll discuss shortly. In general, if you can represent your design using 1149.1-2013, you can also use 1687.

The other assumption is that standard 1149.1 is required to provide the JTAG TAP and state machine, as well as all boundary scan and board test functionality. Today, both standards are based on having the internal TAP available. However, in the future, 1687 could be modified slightly to support other chip interfaces such as I2C.

There are several factors that may affect which standard is best for your specific needs. We'll discuss a few of the bigger ones in this article:

Legacy IP – (IP that has already been designed and in use)

One point of controversy might be the IP that will go into your design. Internal IP or 3rd party IP has the same issues. By the strict letter of the standard, 1149.1-2013 says the TDR register segment that accesses the IP has to be defined in the IP. Does that mean you can not use 1149.1

if your legacy IP does not contain a TDR? Not really, simply develop your PDL for the instrument with a wrapper around the IP that contains the TDR segment bits. However, the Chip Integrator will need to insert that same wrapper structure or equivalent to be compliant to the standard. 1687 does not have a requirement of the TDR segment being with the IP. However, it does have another language and additional work that is left to the Chip Integrator to provide access to the IP.

Legacy Chips – (Existing designs that have JTAG and IP but can not be modified)

You may already have a design in production and want to utilize one of these new standards. This is a pretty cut and dry process to figure out if you can utilize either of these standards. Basically if your design uses JTAG via an instruction and a data register to control your IP, you are a good candidate.

You can probably use either standard if your IP connects directly to the TDR. If there is logic between them, 1687 is the way to go using ICL to describe the logic, IP, and TDR. Advocates of 1149.1-2013 could argue that you can write PDL for the IP to include the IP, the TDR, and all the logic in between. Depending on the logic, developing the necessary patterns may be more difficult than using 1687.

IP Providers

We have a large part of our industry developing and supporting reusable IP. This includes both internal development groups within a company and 3rd party companies. A lot of IP providers have already started to look into supporting 1687 (PDL and ICL) and 1149.1-2013 (PDL and BSDL packages). We believe the marketplace will require IP providers to support both standards in the future. If they don’t, the Chip Integrator may find another IP provider that does.

Both of these standards can enable IP Providers to have more control over how their IP is accessed for test. This allows better management of how customers may integrate the IP to avoid problems, thus reducing support issues and risk while simplifying chip integration of the IP and improving quality.

KISS Principle – (Keep It Simple Stupid)

This old adage has never been used enough in our industry. Why make things more complicated if you don’t need to? If a straightforward network can work, why make it complicated regardless of which standard you end up using? If your company has an established flow using 1149.1 and you want to add a straight forward IP flow, you may want to consider starting with 1149.1-2013. This would avoid learning a new language, ICL, and investigating another standard. As we stated earlier, 1149.1-2013 networks are a subset of 1687. So you can always “upgrade” to 1687 if you need to.

New Designs that also need JTAG for Board Test

You may have a totally new design or SOC coming down the pipe and wondering if you should support one of these standards and which one to choose. On a new design or incremental improvement (spin) of an existing product, you’ll have the flexibility to make tweaks in the test structures. Again, why not start with a straight forward approach and have direct IP connections to a TDR segment if possible? This gives you the most flexibility to select either standard.

New Designs that do NOT need JTAG for Board Test

Many existing and new designs simply do not require JTAG to be included to support board test. If this is your situation, there are a few simple things you can do to still get most of the benefits of the new standards. Remember, the standard JTAG interface requires at least 4 dedicated pins to be compliant. The key word here is “compliant”. We always suggest being compliant to the standards whenever possible. Therefore, if your design can add 4 dedicated primary pins, then simply add 1149.1.

However, if you don’t have 4 pins available and are not required to be compliant by your customer or board manufacturer, there might be a middle ground.

- insert a minimal JTAG controller without a Boundary Scan Register (BSR)
- instead of 4 dedicated pins, use 4 shared pins in test mode
- use the TAP controller for your chip test mode controller

The logic required is very small to support this structure and will still let you take advantage of utilizing 1687 or a subset of 1149.1-2013 for IP testing. Again, today both standards require 1149.1 as the interface. However, 1687 could be simply modified to support other interfaces such as I2C and probably will be in the future. Remember however, by the letter of the standards, neither approach will be compliant. Non-compliant designs may have a issues with board test, field test, or system testing. Vendors and partners downstream have tools that rely on compliance.

Differences in the standards

1149.1-2013 has many new features with pre-defined support in addition to support of embedded IP. The other major sections in 1149.1-2013 that have been added are:
1) Component Initialization support
2) Test Mode Persistence
3) Segmented Boundary Scan Register
4) Register Constraints
5) Power Domain Support
6) Reset Control
7) System Clock Definition
8) Electronic Chip ID (ECID)

IP Support and Methodology

1149.1-2013 supports a few simple element blocks or structures to enable a network to be constructed. Conceptually, this type of approach is generally simple to understand and fairly straight forward to implement while still being able to generate complex networks.

1687 can support more complicated or custom networks if required. While this can give the Chip Integrator more flexibility it adds some additional steps to fully define the network.

Other than a few “gotchas” that we will outline below, both standards can represent most of the designs of today. Your company’s specific design flow and methodology may indicate which one makes sense for you to follow. If you’d like to give the Chip Integrator more flexibility to define the network with greater detail, 1687 is a good way to go. Don’t forget, with this flexibility also comes the additional work of defining and verifying things are correct. A lot of the insertion, verification, and checking of the network can be automated but it is still up to the Chip Integrator to make sure all is functional. It could also be argued that 1149.1-2013, because it limits the set of building blocks, is more straight forward to integrate and is easier to understand conceptually.

However, if you want to make the Chip Integration easier and give up a little flexibility, either standard can work by including the TDR segment in the IP.

A BIG question that needs to be answered is, “Where is the PDL written to?”

If your PDL is written to a TDR segment at the edge or part of the IP, either standard can work. If your PDL is written to ports of the IP, then 1687 is a better route. You could take that PDL and rewrite it to compensate for the effect of any intervening logic and include the TDR segment to be compliant to 1149.1-2013. However, depending on your flow and the complexity of logic, this could be very difficult. Again, why make things difficult?

Commonly Used Terms for Network Building Blocks

This is a very high level conceptual description of these terms. There are differences and variations on each of these elements that are defined in the corresponding standards. Please refer to each standard for details and usage.

- **1149.1-2013 BSDL Building Blocks**

  SELECTMUX - A BSDL keyword in the register assembly attribute that defines the start of a one-of-N selection structure. It is followed by a list of selectable register segments. The actual MUX is inferred at the end of the list. Also, the **SELECTFIELDS** and **SELECTVALUES** keywords of the structure define the selection register field and selection criteria. The selection register field may be anywhere in the TDR structure except inside one of the segments selected by this selection register field.

SEGSEL/SEGSTART/SEGMUX - Logic blocks defined in the 1149.1 Standard Package and used to control and delimit an excludable (bypassable) TDR segment. If the SEGSEL cell immediately precedes the segment, then it delimits the start of the segment (scan-in); otherwise, a SEGSTART delimits the start. The SEGMUX always delimits the end of the segment (scan-out). SEGSEL excludable segments can be nested within other excludable or selectable segments. The SEGSEL cell may be located or duplicated in other TDR structures such as in a user-defined register or an init_data register. The combination of SEGSEL immediately prior to the segment and SEGMUX is analogous to a 1687 Segment Insertion Bit (SIB with post-mux), see definition below.

DOMCTRL – A single shift-register bit with an update cell that is used with a power domain controller and a SEGSEL bit to ensure that a power domain is powered up before an excludable segment in that power domain is included.

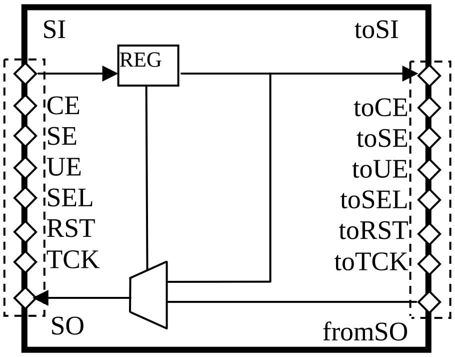
- **1687 ICL Building Blocks**

  ScanMux – A multiplexer than can select one-of-N paths on a network.
  ScanRegister – A single or multi-bit register that defines a TDR or TDR segment which can have a control and/or update cell.
  Additional Blocks - There are other blocks that are defined by the 1687 standard to add flexibility and support for advanced structures:
    DataRegister and DataMux - Analogous to ScanRegister and ScanMux, but only used in the parallel datapath, not the scan path. These would be used to define registers and multiplexer in the scan path control logic or instrument control logic.
    ClockMux – Multiplexer specifically for clock signals.
    LogicSignal – Used to define a cloud of logic for network control.
    OneHotScanGroup and **OneHotDataGroup** - Multiplexers based on one-hot control signals to be used correspondingly in scan path and data path.

- **Segment Insertion Bit (SIB) :** A commonly used term in 1687. It should be noted that a SIB is not actually a cell defined in the ICL language, but is a term used to describe a combination of a register and a multiplexer for network control. A SIB includes a single bit register (including an update stage) on a scan chain which controls a multiplexer to select or bypass a segment of that chain in the active shift path.

Pre and Post-Mux SIBs
The terms "pre" and "post" refer to the position of the multiplexer within the SIB design. 1687 and 1149.1-2013 can support both SIB structures.

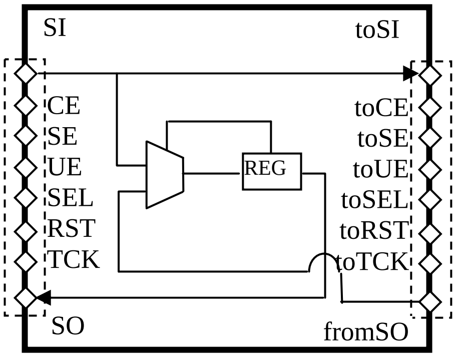
A post-mux SIB places the multiplexer after the control and IP bits in the chain.



SIB with Post-Mux

A pre-mux SIB places the multiplexer before the control bit in the chain

The other type of SIB implementation could be a distributed SIB. In this more custom SIB, the control bit is somewhere else in the chain or in another TDR.



SIB with Pre-Mux

Architectural or Circuit Considerations

As we stated earlier, there are very few types of features that can’t be supported by both standards. There are a few that we think deserve to be noted.

1) **Support for Multiple Embedded TAP controllers within a design**
1149.1-2013 does not address this issue. IEEE Std. 1149.7 does address embedded TAPs but is not being adopted widely. Unfortunately the 1149.7 specification is very complicated, which is leading the majority of companies away from supporting it. 1687 does support embedded TAPs by utilizing the ICL language to describe these TAPs. However, this may be a complex issue to handle in 1687 too.

2) **TDR Constraints**
1149.1-2013 can support constraints. This enables a way to state that one or more specific TDR fields cannot have a value resulting in setting a prohibited or non-recommended combination of values. The constraint to be checked can be very complex, with a full set of Verilog-style logical and arithmetic operators available. The tool used should automatically check prior to each scan to make sure constraints are not violated. 1687 does not have this feature.

3) **Network initialization and reset**
1149.1-2013 defines rules for what the state of the network is after a reset. Since 1149.1-2013 has pre-defined network elements, it also defines rules for the reset state and type of reset for these elements. The 1687 ICL language can define what these values are so they can have more flexibility during the integration phase. Both standards can also generate a local reset for a specific IP or instrument.

Comparing Common Features

The table below shows some of the features of both standards to help compare and contrast their usages. This is just a subset of major features we thought deserve mention. As you can see, there is a lot of overlap between these standards; however, the actual implementation of these features is different within each standard.

	1149.1-2013	1687
Standard Chip pin interface	yes ✓	no ✗
Supports Board Test with BSR	yes ✓	no ✗
Supports Power Domains easily	yes ✓	no ✗
Supports Constraints	yes ✓	no ✗
Automation with 3rd party tools	yes ✓	yes ✓
Support for IEEE 1500	yes ✓	yes ✓
Complex networks	yes ✓	yes ✓
Heirarchy in networks N deep	yes ✓	yes ✓
Ability to isolate instruments	yes ✓	yes ✓
Daisy chain or hierachical IP networks	yes ✓	yes ✓
Customizable Network resets	no ✗	yes ✓
Random and complex gates within a network	no ✗	yes ✓
Multiple embedded TAPs	no ✗	yes ✓
Enhanced modeling of functional clocks to IP	no ✗	yes ✓
May allow simplified sub-module verification	no ✗	yes ✓
Security	no ✗	no ✗

** The above table refers to straight forward implementation of the features listed. Advocates of a standard may be able to find work-arounds for some of these features.*

Some recommendations and thoughts

As stated earlier, it may be some time before the market sorts out the optimal use of these two standards. In the meantime, there is already a lot of mis-information out there. Having test experts on both standards working groups, we would like to offer a few suggestions.

- IP suppliers should support both standards. Supporting only one will exclude an IP provider from a segment of the market. It is highly probable that many SOC type chips will have to use and support both standards. Supporting both standard's file formats will only add a small incremental effort over just supporting one.

- If the IP contains I/O that has to be initialized prior to use (including board test), then the 1149.1 standard initialization instructions will need to be supported. This includes an init_data register segment, preferably in the IP. In general, IP containing chip level I/O may be more naturally described using 1149.1.

- Whenever possible, build the necessary test data register segments into the IP. If this is a new design, design in the test data register segment. If it is an existing design with registered inputs and outputs, convert the parallel registers on the inputs and outputs to 1149.1 scannable registers (no MUX required) the next time the IP is modified or updated for any other reason. If this is an existing design without registered inputs and outputs, consider providing an 1149.1 compatible wrapper (hard or soft) with the IP to provide register access. The wrapper could be a simple manually coded register segment, or it could be an automatically generated 1500 wrapper, especially if other capabilities of the 1500 wrapper could be utilized. The IP provider only needs to do this once, eliminating a source of possible integration errors, as opposed to every customer having to do it for themselves.

Once the test data register is available, both standards can be supported. The IP supplier then controls the description and capabilities of all control and status fields, and can supply named values to be written to or read from the register fields. PDL for both standards can be written using register field names and named values. This should eliminate a layer of potential integration errors, reduce support requirements, and give the IP supplier much better control over the integration and use of their IP.

Many consider it an advantage that they only need to supply the PDL procedures for an IP without the test data register segment included. Certainly, 1687 PDL procedures written to the ports of an IP offer an advance in capabilities. However, the IP supplier loses the ability to define the interface to the TDR. This may make more work for each integration team, and is forgoing the opportunity to eliminate a layer of possible errors in the ICL generation. These possible errors may require the IP supplier's assistance to identify and correct.

- For existing IP without a test data register built-in, definitely provide 1687 PDL and IP level ICL for any and all public capabilities, including functional and test features.

- Retargeting PDL to chip level JTAG sequences – Retargeting is a commonly used term that is a bit misleading. It gives people and companies the impression this is a reformatting issue. This step is more than just reformatting. An understanding of JTAG sequences and the state machine is required. An analysis of the network is also required to understand the most effective means to apply patterns to reduce cycle count. This network analysis requires an algorithmic approach to optimize dynamic switching of the network. Just to do simple retargeting, without even considering any optimization, requires JTAG understanding AND analysis of the network. For optimization, even more analysis would be required of the above along with resource constraint analysis.

  PDL retargeting should ideally only occur once in the flow. Different tools will very likely retarget patterns slightly differently. Optimization of network switching sequences and interpretation of some PDL commands such as iApply and iMerge can lead to different pattern sets.

  The retargeted chip level patterns will then be used in the simulation/verification phase. These verified patterns should then be reformatted to downstream formats such as ATE testing, board level testing, or system level testing. This enables debug and diagnosis if a failure occurs from the test system back to design.

If it is necessary to retarget the chip level PDL to other downstream formats using another retargeting tool, these new unverified patterns should also be simulated to verify correct operation.

- 1687 or 1149.1-2013 Tool Providers
  As we have pointed out in a few places, the development of tools around these new standards can vary greatly. Selecting a tool and a provider is similar to selecting a surgeon. Select a specialist that understands your issue and has been involved in that field for many years. Understanding and doing internal chip IP design, simulation, pattern generation, and JTAG is essential to getting an optimized, high quality solution with these standards. This is not a pattern conversion problem. It is an internal chip development issue for IP developers, Chip Integrators, Chip verification, Chip test or FA.

- Security of IP
  Protection of internal functions is always an important topic when talking about embedded Intellectual Property. Neither 1149.1-2013 nor 1687 explicitly address security. However, after you understand the basics of how to generate a network with one or both of these standards; you can start to imagine how to obscure access to an IP. Certainly, there can be several layers of different types of security depending upon the importance and cost of the IP. Supporting one or both of these standards may enable you to develop a standard security method to your required level. Security will be a topic we will touch on in a later article.

Conclusions

Clearly both 1687 and 1149.1-2013 have a lot of common advantages in efficiency, reuse, automation, and improved quality that affects several stages in the development of a chip or SOC.

We have tried to describe the main differences between the two standards and which situations might make one a little more advantageous over another. However, in most cases with a little planning, most chips could use either standard successfully. Remember that 1149.1-2013 IP support is mostly a subset of what 1687 can do. If your situation allows you to select either, then the tradeoff of basic flow with limited capabilities versus a more flexible flow maybe the question.

One of the larger advantages of 1687 is the ability during the Chip Integration phase to define more complex structures in the network that can be described in ICL. One of the larger advantages of 1149.1-2013 for instruments is simplicity in only needing one standard. It may be more intuitive and straight forward to understand the idea of simple blocks to build a network versus learning and supporting the ICL language.

A large IP business advantage may accrue from supporting both standards for embedded IP access and testing. In the future, there is nothing prohibiting an SOC from having both standards within the same chip. Certainly, 1149.1 will be required if board test support is required.

Both standards enable a huge amount of reuse, improved quality, reduced time to market, and automation. Both standards can provide a means of standardization of IP insertion, IP verification, internal IP test interface, common ATE and Board hardware interfaces, and more.

Full Disclosure About SiliconAid Solutions Inc.

SiliconAid has contributing members in the following IEEE working groups: 1149.1-2013, 1687, and IEEE 1149.6. SiliconAid provides a full suite of chip focused products to support 1687, 1149.1, and 1149.6 standards. (IE. We are not a board test focused company. We are a JTAG based standards chip focused company.)

Want to find more information on SiliconAid Solutions 1149.1-2013 and 1687 tool suite? (www.siliconaid.com/SAJE_art3.html)

SiliconAid Solutions, Inc. is recognized as a leading provider of Design for Test consulting services encompassing methods, implementation, as well as team augmentation and training. The Senior DFT Consulting Group has been recognized for comprehensive support of all standard EDA DFT solutions.

SiliconAid also delivers world class **chip-focused** JTAG software solutions to validate, verify, and utilize IEEE 1149.X and 1687 related industry standards. The exhaustive chip-level validation, verification, and debug of IEEE JTAG-compliant implementations is based on over 20 years of testing and thousands of designs from multiple satisfied worldwide semiconductor companies. SiliconAid has corporate headquarters located in Austin, Texas and was founded in 2001.