IJTAG Compatibility with Legacy Designs - No Hardware Changes

By: Al Crouch, Jim Johnson, Bill Atwell

Overview
By now you have heard the buzz in our industry about the new IJTAG standards (IEEE 1687 and 1149.1-2013) that have come out in the past year or two.

This is the first of a series of articles on incremental adoption of IJTAG. The first question you should ask is, “Why do I need this IJTAG stuff”. We already have an old method that we use for Embedded IP testing. Why do I need the headache of trying to use a new standard and a new flow? Or maybe you have heard, “This is only test, we can add it in later after tape-out, no big deal.”

From a practical point of view, if a new method or standard doesn’t save you money, why do it? Remember that money takes many forms in our industry: time saved, reduced work, improved quality, software automation, improved efficiency of cross functional or cross regional teams, improved debug, faster time-to-market, faster time-to-volume, and many more marketing things you have probably heard about IJTAG. The interesting thing about IJTAG is that most of the marketing hype you hear is really true.

IJTAG allows you to “Plug-and-Play” embedded IP into most chips and to automate the test insertion, verification, and pattern generation, while improving quality. It does sound too good to be true. Another benefit of IJTAG, that you may not recognize at first glance, is that we now have a standard way for IP providers, chip developers, board manufacturing, and system people to document and describe how a specific embedded IP should be connected and operated. This type of Eco-System definition has many benefits.

As you already know too well, SOC chip design uses more and more embedded IP from more and various sources every year thus requiring more consideration of reuse and portability for efficiency. This trend is not going to change and will only be accelerated as chips get more complex in the future. That’s a critical reason supporting why you need IJTAG!

The question for this article is, “Can I retro-fit IJTAG on an existing Chip and Why do I want to?”

Since the chip design has already been completed in this scenario, we can’t save chip development time or time-to-market like on the new design. However, there are many other benefits that can be leveraged.

Typically, the IP provider’s original verification suite can’t be reused because it is driving the ports of the IP directly. The SOC chip integrator needed to generate tests after it was integrated, since direct driving of the ports was not possible after the IP was integrated in the SOC. This is a manual human effort and errors could be injected. It is also difficult to tell exactly how well the embedded IP is being exercised and tested after it has been integrated into the SOC. Ideally, the original IP provider’s patterns would be applied since the IP developer understands the IP functions the best. One benefit of IJTAG is that it allows the reuse of the IP providers patterns. This also gives an ownership path back to the IP provider in the event test escapes are found by the end customer.

Developing and learning IJTAG on an existing design has many benefits. One of the main benefits is that it allows embedded IP to become fully portable with reusable vectors. It also allows you to not be in a critical path as you would be with new chip development. This provides a proof-of-concept and can demonstrate real benefits to management which can reduce risk and improve buy-in for the “decision makers”.

For additional articles on IJTAG or SiliconAid IJTAG products, see the Tech Corner at ***CLICK HERE***.

Why not use your legacy design to get up the IJTAG learning curve? Get a flow established and demonstrate to management the value of IJTAG going forward. Develop an IJTAG library of embedded IP that can be used on any chip in the future.

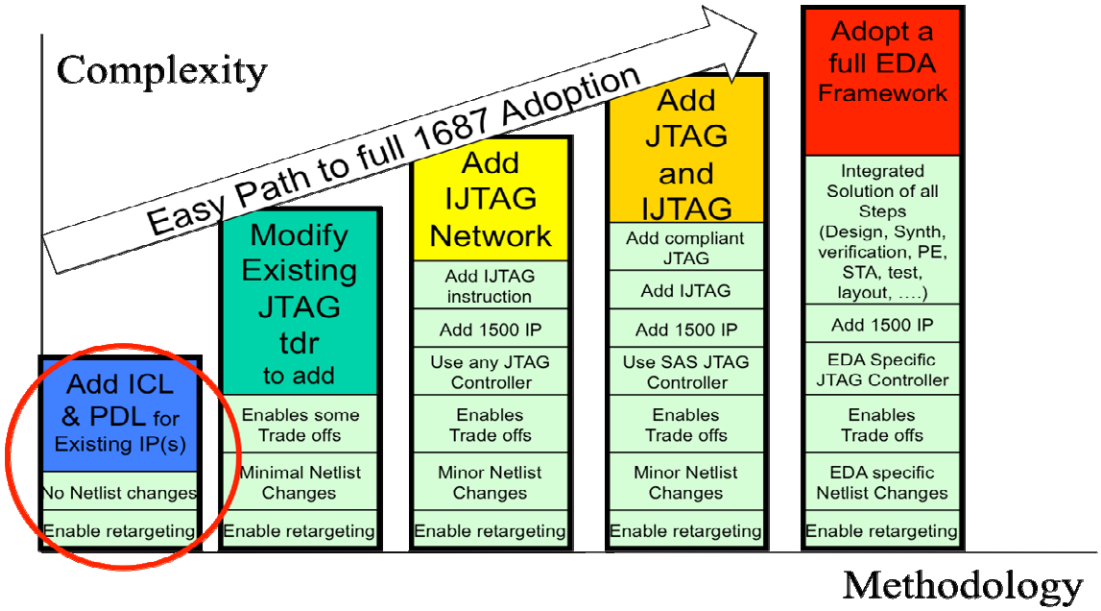

Figure A: IJTAG Adoption Stages from least effort to most effort

After a chip is retro-fitted into IJTAG, there are several other benefits for future work that become possible. The embedded IP’s used in that chip now have the files required so they can be “Plug-and-Play” into any future chip. This library type view development is done once for a specific IP. Additional quality improvement patterns can be developed easily and incrementally added. The chip can now claim to be IJTAG compatible and may win additional design-ins for different end customers.

Another main justification for converting a design from ordinary compliant JTAG to an IJTAG design is to provide tests and test procedures with the IP and/or IC and to reuse them later. Reuse means taking tests made during IP verification and/or IC design phase and allow them to be used again to automatically create verification and test patterns for chip, board, or system needs downstream.

Introduction

We get lots of questions from customers interested in IJTAG. It has a lot of advantages but how can we use it today?

Many customers don’t realize that they may be able to take advantage and leverage IJTAG on existing legacy designs with no changes to the chip.

You can use the IEEE 1687 Standard or IEEE 1149.1-2013 on older IEEE 1149.1 compatible designs, without changing your hardware, and can therefore reap many of the advantages of IJTAG without changing your design or design tools. This exploration includes designs that are purely IEEE 1149.1 or designs that also incorporate IEEE 1500 Wrappers.

IEEE 1687-2014 is a recently ratified IEEE standard that defines access, operation, and management of embedded instruments within chip designs. It is a standard that is complementary to IEEE 1149.1 and IEEE 1500 standards. It adds documentation and hardware techniques to existing standards-based designs, to enable embedded instruments to be reusable and portable.

IEEE 1149.1-2013 has many new features, including the support for basic IJTAG. It has most of the same basic benefits as IEEE 1687. The connections to the embedded IP are described in BSDL (Boundary Scan Description Language in IEEE 1149.1-2013) extensions and thus there is no requirement for ICL (Instrument Connectivity Language in IEEE 1687). However, the lack of ICL also limits some of the IJTAG networks that IEEE 1149.1-2013 can support.

The path to adoption of IJTAG can be incremental. As Figure A shows, use of a legacy design does not require changes to the chip design. You just need to create the files required to support IJTAG and take advantage of automatic simulation and pattern generation. *(shown in blue)* Any JTAG controller should be able to be reused and supported.

Legacy Designs that could be compatible with IJTAG

To be realistic, not every existing design is easily retrofitted to IJTAG. However, there is a significant subset of legacy designs in the world today that are compatible with IJTAG. A common test access method companies have used for embedded IP for many years, and still today, is the widely adopted JTAG (IEEE 1149.1). Where private or user instructions are added to the JTAG TAP controller’s instruction register that allow test data registers (TDRs) to interface to various embedded instruments. Generally, an instruction will select a particular TDR and that TDR will interface with an embedded IP. Chips with this type of approach are IJTAG compatible and will be the focus of this article.

Basically if your design uses JTAG via an instruction and a data register to control and/or observe your embedded IP, you are a good candidate for IJTAG.

Basic Requirements for being IJTAG compatible with a Legacy Design:

- JTAG 4 or 5 pin interface
- JTAG compatible TAP controller
- Embedded IP connected to user or private JTAG TDRs (Test Data Registers)
- Embedded IP with IEEE 1500 Interfaced to JTAG TDRs (optional)

The original embedded IP flow probably had many manual steps for integration, verification, pattern generation, and board test.

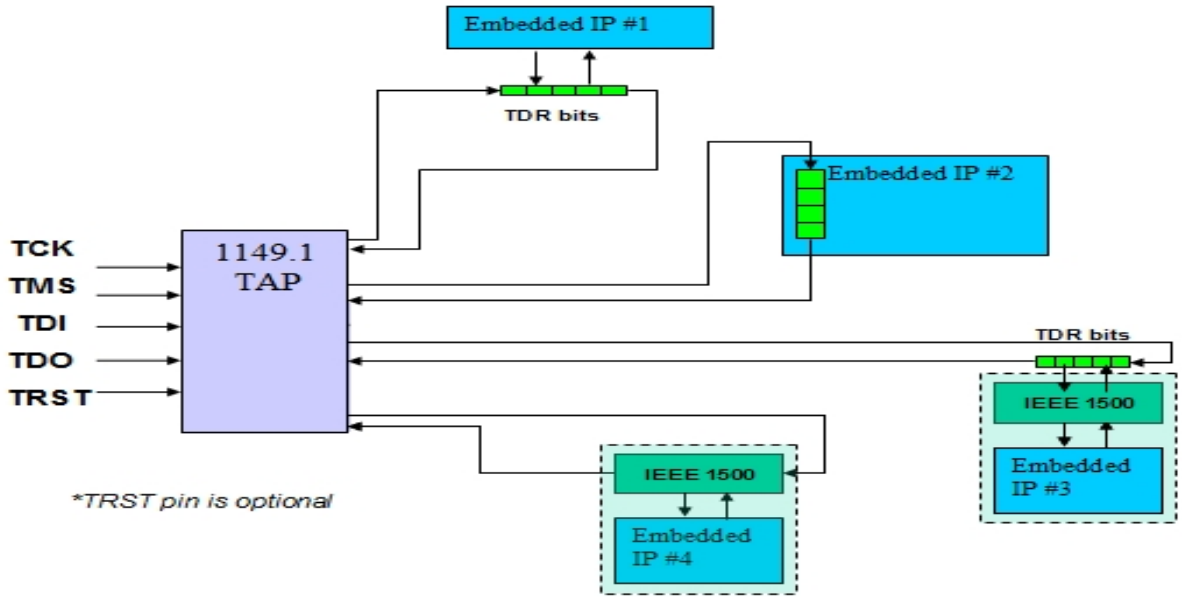

Figure B: Example 1149.1 instruction-accessed structures

Figure B shows examples of IEEE 1149.1 User or Private instructions accessing embedded IP. Any one or combination of these types of structures can be IJTAG compatible. Each instruction accesses a different Embedded IP. Embedded IP#1 shows an IP with a separate TDR segment. Embedded IP#2 shows an Embedded IP with the TDR segment built into the IP. Embedded IP#3 shows an embedded IP with an IEEE 1500 Interface and a separate TDR segment. Embedded IP#4 has an IEEE 1500 interface that can also represent the TDR segment for that instruction.

You may have many more Private or User instructions and many more embedded IPs with any or all of these IP configurations or similar structures. The types of embedded IP can also be daisy chained together on one instruction and still be IJTAG compatible.

Many EDA vendors and IP providers provide IP with 1500 or serial access ports. Some IP vendors are also starting to support the IJTAG files (ICL and PDL) required for the IP.

Maybe your design has combined JTAG with a TDR segment connected to the embedded IP that has an IEEE 1500 interface, keep reading!

IJTAG would allow you to automate these steps on your existing legacy design. IJTAG would also allow reuse of the IP patterns for any other chip.

You get the advantages of reducing risk, reducing schedule, improving quality, and improving reuse all at the same time!

After an IJTAG flow is developed, adding additional tests to improve quality for the embedded IP becomes much easier! A high level language called PDL (Procedure Description Language) is now available to create new tests. The engineer can now create a pattern for the embedded IP without needing to know or understand JTAG or how to enable the path to the IP. He or she would simply use high-level commands to read or write to the ports of the IP with PDL iRead and iWrite commands.

What files need to be created
One of the differentiators between basic JTAG and an IJTAG implementation is the network description of the IJTAG network and PDL. The IEEE 1687 ICL or the IEEE 1149.1-2013 BSDL extensions represents the pathway between the instrument and the chip-level test access mechanism. The PDL represents the operation procedures or vectors associated with the instrument directly. The question remains, however, where do these files come from, who creates them, and how are they created?

The main requirements needed to support an IJTAG flow is the addition of ICL and PDL at the IP level and chip level.

What is PDL?

The Procedure Description Language, most commonly called PDL, are procedures for stimulating and observing data *(patterns)* to an embedded IP. PDL is a language that is associated with the interface of an embedded IP. In order to make the PDL reusable, there must be a description of the pathway to get those vectors to the pins of the finalized IC. That is where ICL comes in.

What is ICL?

The Instrument Connectivity Language, most commonly called ICL (pronounced ick-el), is a modeling language that describes the connections or access network between the JTAG TAP controller and the embedded IP.

Figure C shows a JTAG example of what needs to be described to conduct IJTAG's PDL retargeting process. A simple memory BIST IP is shown that includes two input signals (START, RESET), two output signals (DONE, FAIL) and their connection to a JTAG TDR. All of these IP connections will be described with simple ICL. Note that the PDL is referenced to either the IP signals or to the TDR itself. Additionally, the TDR's connections to the JTAG's serial TDI and TDO path must be described in ICL, but the TAP and TAP Controller are described with JTAG BSDL. The one simple BSDL change needed to support IJTAG is the addition of the key phrase AccessLink to relate the JTAG instruction(s) to the IJTAG network in IEEE 1687. In the case of 1149.1-2013, ICL is not required and the network descriptions are in the BSDL as extensions.

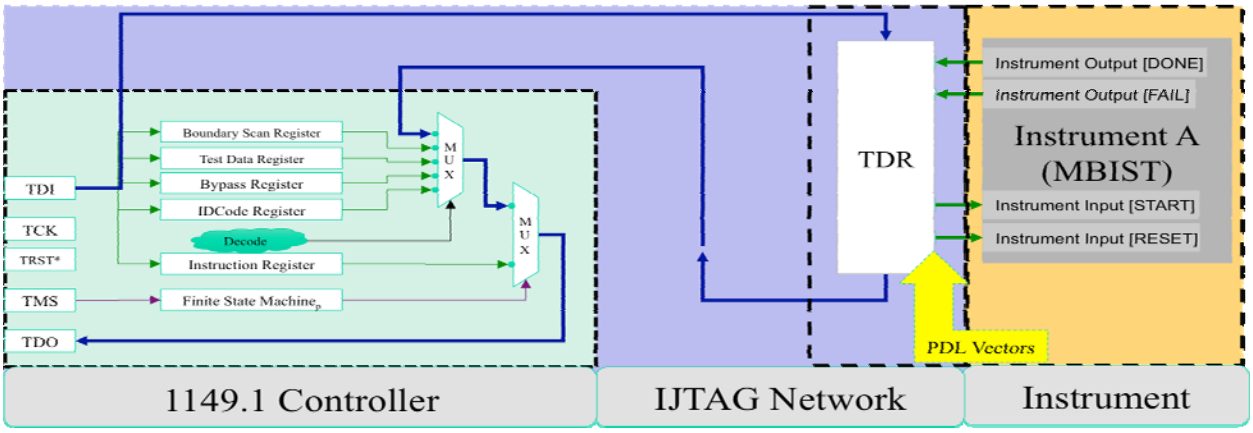

Figure C: Example of a simple JTAG accessed instrument

How to create ICL and PDL for instruments

Ideally, as IJTAG becomes adopted more widely, ICL and PDL for the IP will come from the IP developer. Today, chip design or test engineers need to generate these files. These files can be manually generated since PDL and ICL are simple higher-level languages. However, this manual method can be time consuming and error-prone depending on the complexity of the

patterns.

In many cases, the verification simulation from the IP developer might be able to be converted into a PDL with the use of a Verilog API or other means. The signal interface to the IP will need to be described in ICL. This could be possible be extracted from the Verilog.

SiliconAid has developed software IP that can automatically generate the PDL and ICL from the simulation environment provided by the IP supplier. For additional information on generating IP level ICL and PDL, ***CLICK HERE***.

Creating Reusable IJTAG Friendly Embedded IP

The ICL and PDL can be delivered as files with the embedded IP. In this library type form, the files may be reused for verification when the instrument is integrated into the overall chip. If the embedded IP is not modified (so the vectors and interface description are still valid), then the ICL and PDL files may be reused at the chip-level for IC test and debug, board level integration, and test and debug when the final chip is used in a product or system.

Creating IJTAG Friendly Chips

Typically, an IJTAG automated insertion tool is used to create the IJTAG network to and from the IP, then the ICL can be automatically generated.

In our scenario of an existing chip, all the TDR to IP connections have already been done. However, you can duplicate the connections in an IJTAG insertion flow to generate the chip ICL. You can always generate the ICL manually. This may not be difficult but depends on the number of IP.

To reuse the IP level PDL at the chip-level, an IJTAG network analyzer and retargeting tool must be brought to bear. For additional information on IJTAG network and retargeting software, ***CLICK HERE***.

Delivering Chip patterns to ATE and Board Test

When the chip IJTAG flow is complete, several outputs to supply ATE and Board test can be automatically generated. These may only require a subset of chip level patterns. It is up to the chip provider to automatically generate and package chip-level vector formats such as SVF or STIL, and deliver that set of patterns to the end user.

Conclusions

There are many advantages of using an existing design that is compatible with JTAG and retro-fit it into an IJTAG flow. In many cases this can be done fairly quickly and has many benefits with very low risk.

Some of the major barriers to any new flow are the learning curve and the buy-in from management. After the flow is proven, additional IP can be added for future new chip development and existing design elements can be considered as portable and reusable for future designs.

As stated earlier, this type of retro-fitting approach can be accomplished with either IEEE 1149.1-2013 or in IEEE 1687 standard.

For additional information please visit http://www.siliconaid.com