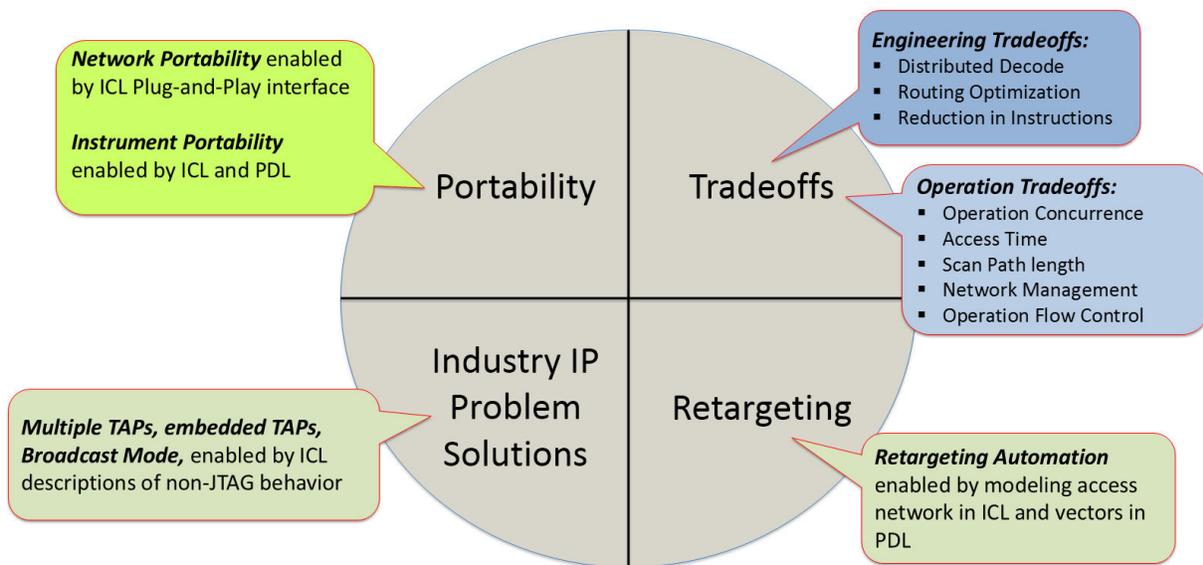


IEEE 1687 IJTAG |

THE FUTURE OF EMBEDDED INSTRUMENTS

IJTAG Capabilities



BY AL CROUCH



Al Crouch – Chief Technologist, Embedded Instrumentation Methodologies and IJTAG

Al investigates the use of embedded instruments for IC test and debug, board test and debug, and software debug. He is a Senior Member of the IEEE and serves as the vice chairman of the IEEE P1687 IJTAG working group that is developing this standard for embedded instruments. He has contributed significantly to its hardware architecture definition. Al is also a member of the P1838 Working Group on 3D test and debug, and co-chair of the iNEMI BIST group, which is defining the use of embedded instruments for board test. Al's previous experience includes design-for-test and debug at various semiconductor companies, including TI, DEC and Motorola, as well as chief scientist at startup companies DAFCA and INOVYS.

Table of Contents

Executive Summary	5
Why IJTAG was developed	7
Addressing those growing pains	8
Instrument Portability and Re-Use	9
Hierarchical Instrument Networks	11
Managing Access Time.....	11
Concurrency and the Run-Test-Idle Requirement	14
Power Domain Management.....	16
Physical Tradeoffs	18
Embedded TAPs	20
Simple and Advanced Concurrent Instrument Operations	23
What’s next for the IJTAG standard?	24
Synchronous Instrument Concurrency.....	26
Future IJTAG Standards Activities.....	27
Conclusions.....	29

Table of Figures

Figure 1: IEEE 1687 IJTAG Capabilities.....	6
Figure 2: IJTAG’s swap operation involving a ScanMux selecting one of two scan paths.	12
Figure 3: A configuration representing IJTAG’s concept of a SIB adding a scan path segment. 12	
Figure 4: An implementation of a SIB and how it adds a scan path segment to the active scan path.....	13
Figure 5: Example of variable length scan paths.	14
Figure 6: A two-instrument network with SIBs to enable scheduling and concurrent operation. 15	
Figure 7: Implementing three SIBs to operate a register in a switchable power domain.	17
Figure 8: A Shift-Capture-Update-Reset TDR instrument interface cell with local decode.	18

Figure 9: Example of locally generated Select signal from a SIB and moving decode closer to an instrument's TDR..... 19

Figure 10: An embedded fully-compliant IEEE 1149.1 JTAG TAP and TAP controller..... 21

Figure 11: A main TAP accessing three eTAPs in a daisy-chain connection. 23

© 2015 ASSET InterTech, Inc.

ASSET, ScanWorks and the ScanWorks logo are registered trademarks, and Arium and SourcePoint are trademarks of ASSET InterTech, Inc. All other trade and service marks are the properties of their respective owners.

Executive Summary

As is the case with most standards, the development of the IEEE 1687 Internal JTAG (IJTAG) standard for embedded instruments began because engineers were grappling with problems that could not be efficiently solved by proprietary means. Previous to IJTAG, chip providers typically created their own custom and proprietary access mechanisms for the instruments embedded in their chips. These access operations usually did not seamlessly extend beyond chip characterization and test to a 'chip-on-a-circuit-board' environment where those same instruments might be applied in board validation and test applications.

The development of the IJTAG standard brought together a broad spectrum of interested parties who wanted to find a shared solution or set of solutions to these problems. IJTAG came about because an increasing number of instruments were being embedded in integrated circuits (IC) and engineers were encountering difficulties dealing with them. Out of these growing pains, several noteworthy capabilities were developed and included in the IJTAG standard.

Following a brief introduction to the IJTAG standard, which describes where the standard came from and what it was intended to accomplish, this eBook focuses on several of the key capabilities enabled by IJTAG (Figure 1). Of course, the standard specifies a wide variety of simple, yet innovative techniques, such as incorporating engineering and operational tradeoffs, and enhancing ease-of-use, but the point is that this technology enables certain key capabilities that will allow the entire industry to deploy and use embedded instruments more effectively and efficiently. The key IJTAG capabilities highlighted in this eBook include embedded instrument portability and re-use, hierarchical instrument networks, physical engineering tradeoffs and embedded test access ports (eTAP). At this point, it should be noted that although the IEEE 1687 IJTAG standard is always applied and incorporated into individual intellectual property (IP) and ICs, the extent of its value goes far beyond just the IC, extending to where IJTAG is deployed in circuit board validation and test environments as well as end user environments.

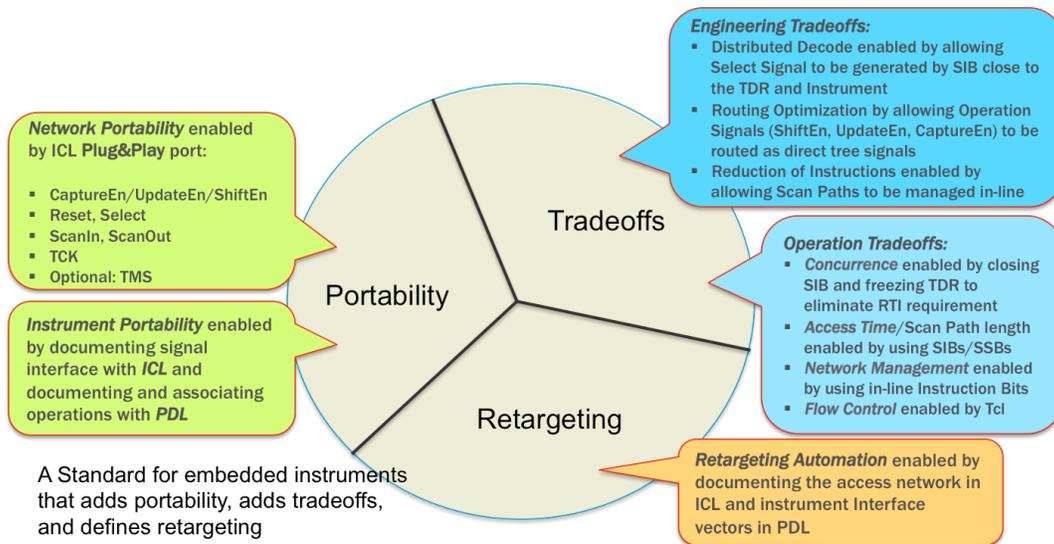


Figure 1: IEEE 1687 IJTAG Capabilities.

But no standard is finished even when the initial version of the specification is ratified and published. Standards expand and grow and evolve to keep pace with the progress of technology in the industry. It is not uncommon for a working group that has developed a new standard to identify other capabilities or functionalities that should be included in subsequent or extension versions of the standard. As an example, consider the evolution of the IEEE 1149.1 boundary-scan (JTAG) standard, which began as the IEEE 1149.1 (dot-one) standard, but was expanded into IEEE 1149.4 (dot-four), IEEE 1149.6 (dot-six), IEEE 1149.7 (dot-seven) and IEEE 1149.8.1 to include capabilities such as analog boundary scan, a two-wire port, mixed-signal test bus, and interconnect testing of passive and active components.

For the IJTAG working group, a key capability that could not be fully implemented in the standard's first version was synchronous instrument concurrency, or the application of multiple instruments simultaneously and in a coordinated fashion. Synchronous instrument concurrency is especially difficult across different types of on-chip instrument networks that are driven by different access controllers, such as IJTAG networks controlled by a JTAG test access port (TAP) and some other sort of instrument network driven by a another controller like a Serial Peripheral Interface (SPI) controller, for example. Enabling synchronous instrument concurrency in the IEEE 1687 IJTAG standard was left for a future new version of the standard.

Why IJTAG was developed

The IEEE 1687 Internal JTAG (IJTAG) standard for embedded instruments was developed to address problems that many in the electronics industry were experiencing. Some of these problems were the following:

- Many more cores and blocks of embedded intellectual property (IP) were being developed by a widely diverse and growing supply network.
- With the deployment of more built-in-self-test (BIST) functions such as SerDes BIST, memory BIST, PLL BIST, logic BIST, scan compression, DDR BIST and others, test was being internalized inside chips to reduce the cost and complexity of test.
- More debugging functions and environmental and process monitoring functions were being included within chips to reduce time-to-diagnosis and time-to-yield and to allow chips to conduct real-time adjustments and configurations based on temperature, voltage conditions and other environmental factors.
- Engineers needed greater freedom to make certain design tradeoffs involving silicon area, route congestion, power consumption, timing impact and others.
- Engineers needed to be able to implement operational tradeoffs involving test/instrument scheduling, access path length, access time, power budgeting, interacting with on-off power domains and others.
- Allowances and standardized configurations were needed for including additional embedded TAPs when a core or block of IP is delivered with a TAP and TAP controller as defined by the IEEE 1149.1 Boundary-Scan Standard (JTAG), or when TAP controllers are used in IC designs for timing purposes, such as the regeneration of JTAG's capture-shift-update-reset operation signals locally.
- The application of embedded instruments to test, debug, monitor and configure power and clock domains, and to tune high-speed pin interfaces has become one of the most effective ways to debug, validate and test circuit boards.
- Management techniques were needed for longer scan paths.
- Additional mechanisms besides JTAG were needed to access embedded content in chips because many devices do not have a JTAG TAP, but they do have SPI, I2C (Inter-Integrated Circuit) and other minimal pin interfaces for test, debug and configuration.

Concurrent with the IEEE 1687 IJTAG working group, another group was investigating the feasibility of expanding the capabilities of the IEEE 1149.1 JTAG standard, which at the time was the *de facto* access method for instruments embedded in chips. The group that formed the IEEE 1687 IJTAG working group concluded that there would be growth and scalability issues

with JTAG. As it was at the time, JTAG was a bit too rigid and not efficiently scalable to accommodate the requirements of embedded instrumentation and its anticipated future growth.

So, the IEEE 1687 IJTAG working group determined that a standard dedicated to the requirements of embedded instrumentation was needed. The group had many objectives, but the most prominent goals included ensuring the portability of embedded instruments, documenting instrument operations, and providing access and operational tradeoff options associated with the access network. Furthermore, such a standard could take advantage of the serial scan access mechanisms embodied in the IEEE 1149.1 JTAG and IEEE 1500 embedded core test (ECT) standards, both of which employ the capture-shift-update mode of operations defined by the IEEE 1149.1 JTAG standard's finite state machine (FSM) protocol.

The goal of the IEEE 1687 IJTAG working group was to address the growing pains that embedded instrumentation was experiencing. A primary benefit of doing so was to ensure the portability and re-usability of embedded instrument IP. The IEEE 1687 IJTAG working group addressed some of the growing pains listed above with architectural techniques and allowances embodied in the IEEE 1687 IJTAG on-chip network of embedded instruments. The remaining problem areas were addressed by the creation of two languages defined in the IJTAG standard. One of these, the Instrument Connectivity Language (ICL), is an architectural description language that specifies the on-chip network of IJTAG instruments while the other, the Procedure Description Language (PDL), is a procedure or vector description language that defines the procedures, operations or sequences of operations for a particular instrument or portion of the access network.

Addressing those growing pains

IEEE 1687 IJTAG is a comprehensive standard for embedded instrumentation, but several of its features are extremely noteworthy because they enable powerful capabilities not available in any other standard. These noteworthy capabilities address instrument portability and re-use, design and use tradeoffs associated with the access network, hierarchical instrument networks, embedded test access ports (eTAP) and the instrument vector retargeting process.

Instrument Portability and Re-Use

Several key features of the IEEE 1687 IJTAG standard ensure the portability and re-use of embedded instruments. One is a plug-and-play interface to the on-chip IJTAG access network that allows the re-use of not only embedded instruments themselves, but also of entire segments of an IJTAG serial access network that provides access to each individual instrument interface on that segment of the network. Such a network segment could be made up of tens, hundreds or thousands of instruments. This capability makes portions or subsets of the network itself portable and reusable. It also can be applied to wrapped IP blocks. An example of this case would be IP with a wrapper conforming to the IEEE 1500 ECT standard. Although this IP block would not have its own JTAG TAP interface, it would be capable of JTAG-conforming operations and data signals, and supports an interface similar to the IEEE 1687 IJTAG interface. (For example, IEEE 1500's interface signals of CaptureWR, ShiftWR, UpdateWR, ResetN, WRCK, WSI, WSO and SelectWIR are similar to those of the IJTAG interface.)

Another aspect that affects portability is IJTAG's vector or operation language, PDL. It is not so much the nature of the language itself, but rather its location that affects embedded instrumentation portability. Before IJTAG's PDL, vector languages were only applied to the pins of the IC. That is, these languages were all outside-the-chip vector languages. As a result, any minor change to the IC required additional work to change the vector file as well. In contrast, IJTAG's PDL language was developed specifically so that vectors would be associated directly with an embedded instrument and would not require editing or modification throughout the usage life of the embedded instrument. This ensures the portability of the procedures or processes that operate a particular instrument. One goal of the PDL language was that once a PDL vector file was developed and delivered with an embedded instrument, the PDL would not need to be modified for all subsequent deployments of the instrument or for the instrument's various uses. Because of PDL, the operating procedures for an embedded instrument go right along with it without modification when an instrument moves from one design to the next. This is a tremendous time-saver for design and verification engineers as it reduces the amount of work for engineering users who must apply the vectors/procedures in different test and debug environments.

By essentially bundling instrument IP with its operations vectors, IJTAG will minimize the need for an integrator to modify the IP so that it works with the integrator's vectors. This can be very valuable later should the IP malfunction and a claim is lodged against the supplier of the IP. (See: "Don't change that IP!")

Portability is also affected by the description of the pathway that the PDL takes from the edge of the chip. ICL describes this pathway. With both PDL and ICL, vector application can be automated with the IJTAG standard. The standard refers to this as 'retargeting.'

Retargeting automatically translates the PDL through the current active IJTAG access network configuration to the edge of the chip by using the ICL as the access model.

Even if an embedded instrument IP is already associated with an IJTAG access network and delivered to an integrator as such, IJTAG's ICL language allows the entire access network to be easily integrated into a new design in one or several places. ICL was not meant to be a detailed language describing all of the nuances, design decisions and physical assumptions of an access network. Instead, it provides a high-level description of the objects included in an on-chip IJTAG network of instruments, such as test data registers (TDR), a scan path, ScanMuxes as defined in the IJTAG standard, enable signals and several other objects. The IEEE 1687 IJTAG working group's strategy was that only the chip's JTAG Boundary-Scan Description Language (BSDL), ICL, and PDL would be needed to retarget and operate IJTAG embedded instruments and their access network.

Ultimately, this comprehensive approach solves a critical problem of portable IP by addressing the portability and re-use of IJTAG networks and the instrument IP that will populate them. By eliminating the need to modify both the network and instrument IP so that it might be included in

Don't change that IP!

When IP supplied by a third-party fails during integration (simulation or emulation) or silicon evaluation, the integrator will automatically contact the IP supplier to complain. The first question asked by the supplier will be: "Did you modify the IP or the vectors?" If the answer to either part of that question is yes, the supplier of the IP is off the hook for supporting the IP. The moral of the story: avoid the headaches and go with IJTAG instrument IP that bundles the instrument with its PDL vectors so you don't need to alter anything.

a new design, IP providers can be held accountable for the performance and execution of the IP. If IP is modified, the supplier is typically not responsible for supporting it.

Hierarchical Instrument Networks

One of the critical features of IJTAG that enables many engineering tradeoffs is a simple concept called a Segment Insertion Bit (SIB). The working group developed the SIB because it addresses a number of issues, namely: managing long scan paths, accommodating power and clock domains, and allowing many much needed operational and engineering tradeoffs.

Managing Access Time

The simple and most often referenced capability of the IJTAG standard is its ability to affect the configuration of the on-chip network of instruments, which is often referred to as the scan path. (Since the network is a serial access network, it can be viewed as a scan path with active portions and with inactive portions.) Most frequently, managing access times will involve altering the length of that portion of the access scan path that is active at any point in time. With variable-length IJTAG scan paths, typically a portion of the path is ‘active’ while other portions of the network are ‘frozen or ‘inactive.’ Since IEEE 1687 IJTAG defines a serial access network, the entire network of instruments can be viewed as comprised of scan path segments.

Data traversing the IJTAG scan path is able to alter the length of the active scan path.

Two main methods are defined in the IJTAG standard to accomplish variable length scan paths. One is called a ‘swap’ and the other is called an ‘add.’ A swap is the same method used by IEEE 1149.1 boundary-scan/JTAG and IEEE 1500 ECT, except that for IEEE 1687 IJTAG, the source of the swap can be somewhere besides the centralized instruction register. An IJTAG swap is a specific object defined by in the IEEE 1687 standard and is referred to as a ScanMux in the standard (it is also an ICL keyword). A ScanMux in the scan path can select one scan path or another and the IJTAG Select signal for a ScanMux is communicated over the data path. Figure 2 shows a segment swap bit (SSB) generating a Select signal to a ScanMux.

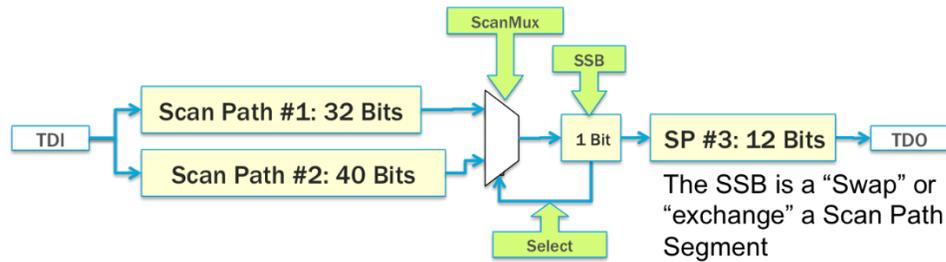


Figure 2: IJTAG's swap operation involving a ScanMux selecting one of two scan paths.

The IJTAG standard places no requirement on the length or configurations of scan paths that can be swapped through a ScanMux. However, at least one path segment must be selected and in the active scan path at all times. One path segment will be designated as the default path following initialization or reset. Note that the selection bit should be in a portion of the scan path that cannot be swapped out of the active scan path. If it were in a path segment that could be swapped out of the active scan path, then the selection bit might not be accessible when the segment has been swapped out.

If one of the swap scan paths is reduced to a length of zero-bits, then the scan path configuration approaches that of a SIB. Figure 3 shows a segment insertion bit generating a Select signal to a ScanMux.

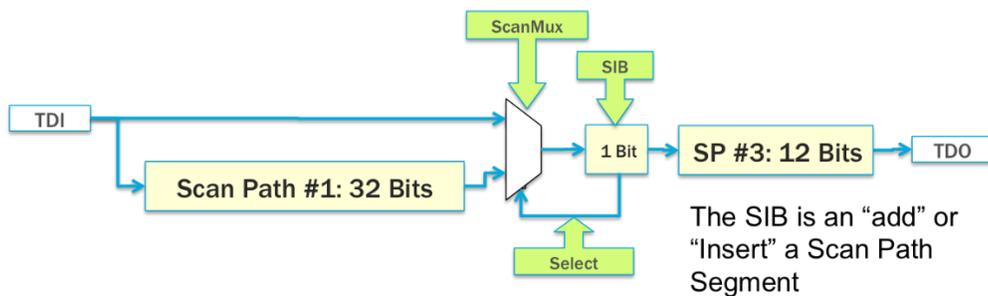


Figure 3: A configuration representing IJTAG's concept of a SIB adding a scan path segment.

Those who are familiar with IEEE 1149.1 boundary-scan/JTAG might see similarities between an IJTAG SIB and the IEEE 1149.1 definition of a one-bit boundary cell, such as a BC_1. These two concepts differ in that a boundary cell is placed across a functional signal, while a SIB is placed across the scan path itself. Also, a SIB's implementation is not mandated to a particular construction. Instead, a SIB is a compact scan path management behavioral construction. In other

both the swap and the add cases, the length of the active scan path is variable, since it can be configured as any of several lengths.

As a result of the SIB concept, embedded instruments (that is, the items interfacing to the TDRs shown in Figure 2, Figure 3 and Figure 4) can be placed on a scan segment that can be brought into the active scan path or the same scan segment can be parked to the side offline from the active scan path. This can be accomplished by transmitting certain data over the scan path. One of the advantages of the SIB's ability to include or exclude scan path segments into the active scan path is the bit length of the active scan path can be adjusted. This translates into another advantage when the bit length of a segment is thought of in terms of the clock frequency on the scan path. Including or excluding a segment will affect access times on the active scan path.

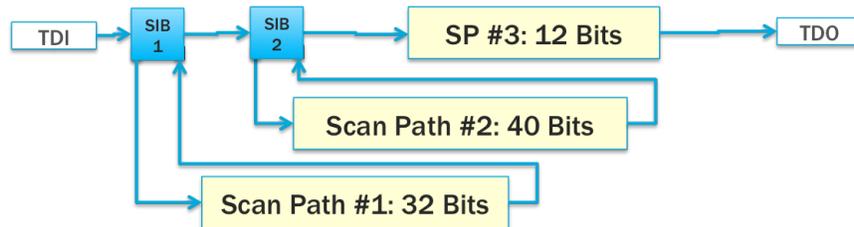


Figure 5: Example of variable length scan paths.

Concurrency and the Run-Test-Idle Requirement

When a SIB is closed, effectively making a segment of an IJTAG network inaccessible, the instruments on this segment of the network are still able to operate and are not forced by the JTAG TAP's FSM into the run-test-idle (RTI) state. While a closed SIB blocks access to the JTAG test data registers (TDR) that interface to embedded instruments on the network segment managed by that SIB, it does not affect access to the rest of the instrument network. That is, the JTAG capture-shift-update operation, which is the access mechanism for the IJTAG network, can continue to operate on other segments of the network (scan chain). A closed SIB will freeze or, technically speaking, become persistent, which would allow any active instrument that is gated by the SIB to continue running and those not running to remain quiescent. Network operation will be directed by the data in the TDR and that data will not change until either the SIB is opened and provides a Select signal so that a shift operation changes the data, or until a reset action occurs.

Defining the SIB was also IJTAG's first step toward allowing concurrent or coordinated instrument operations. For example, an end user might open 10 SIBs, start 10 instruments sequestered behind each of these SIBs and then close the 10 SIBs. The 10 instruments would continue to run while the engineer was using the rest of the IJTAG network to perform other functions.

Figure 6 shows a two-instrument network segment. When it is first accessed, the scan path will be two bits long since only the SIBs will be the active scan path bits. If SIB 1 is opened, it provides access to TDR A. If TDR A were 6 bits long, then the scan path would be 8 bits (6 bits for TDR A, 1 bit for SIB 1 and 1 bit for SIB 2). If SIB 2 were opened, while SIB 1 is closed, and TDR B is 7 bits long, then the scan path would be 9 bits long (7 bits for TDR B, 1 bit for SIB 1 and 1 bit for SIB 2). If both SIBs were open, then both instruments can be operated simultaneously because both TDR A and TDR B are accessible at the same time and the total length of the scan path would be 15 bits (6 bits for TDR A, 7 bits for TDR B, 1 bit for SIB 1 and 1 bit for SIB 2).

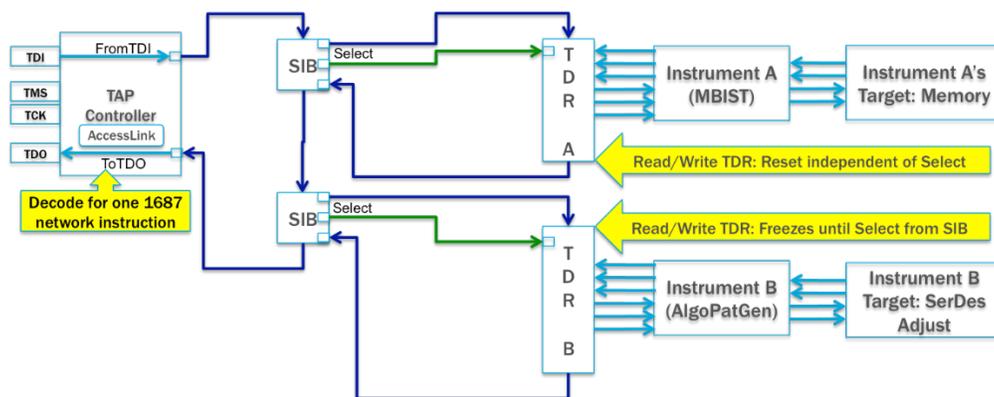


Figure 6: A two-instrument network with SIBs to enable scheduling and concurrent operation.

Now, consider the case of concurrent operation. Unlike JTAG, which does not enable concurrent operations unless they are designed into the hardware, IJTAG supports concurrence and it enables flexible implementations of concurrence. For example, to complete a two-instrument IEEE 1149.1 JTAG operation, an IR-scan would first be performed to access TDR A on Instrument A. This would be followed by a DR-scan that would start Instrument A and the JTAG state machine would park in the RTI state until Instrument A has completed its operation. Instrument A's operation might require 400,000 TCK cycles. After Instrument A has completed

its operation, an IR-scan would be accomplished to access TDR B on Instrument B, followed by a DR-scan to start Instrument B, and another period of time parking the FSM in RTI for possibly another 350,000 clock cycles during which time Instrument B is completing its operation. Instrument concurrence is not possible with this JTAG method and, in this example, the operations of the two instruments would require a minimum of 750,000 TCK clock cycles, without considering all of the IR and DR scans required for setup.

The IEEE 1687 IJTAG method of concurrence would place both instruments mentioned in the example above in a single scan path, but each instrument would be gated by its own SIB (Figure 6). A single IR-scan is required to access the IJTAG network, then DR-scans can open one or the other, or both of the instruments' SIBs. So, if the operation of Instrument A requiring 400,000 TCK cycles is started and the SIB closed (freezing its data state), then the test coordinator could be in RTI for as little as 50,000 TCKs before performing a DR-scan to open SIB B and start Instrument B. This could be followed by closing SIB B to freeze its data state. This means that both instruments run simultaneously and the test is completed in the 400,000 TCK cycles required by Instrument A. If the network were more richly populated with additional instruments, both SIBs A and B could be closed while many other operations are performed simultaneously with Instruments A and B operations. For example, if there were SIBs C through L with an array of temperature sensors, voltage sensors, frequency counters and other instruments, the environment of the chip could be monitored by opening and closing SIBs while Instruments A and B were running.

Power Domain Management

A SIB could also segregate or isolate a TDR and its instrument in a low power area from the active clock/power domain. The alternative would be to maintain the entire clock/power domain in the active state so that the scan path would not be broken. Generally and at a high level, a SIB can isolate or access an instrument in a switchable power domain. However, turning a low power domain on and off for a TDR access with just one SIB is not sufficient. Turning the power to the domain on and off requires a graceful sequence and settling time. In a real-world implementation of a power domain (Figure 7), three SIBs would be needed for a hardware solution. The first SIB closest to the controller would open when asserted and the IJTAG Select signal it generates

would enable the power buffer as well as provide access to another SIB. The second SIB would open when asserted and the Select signal it generates would enable the clock buffer and provide access to a third SIB. The third SIB would provide access to the test data register in the power domain. The time it takes the JTAG state machine to perform a DR-Scan is adequate for the power, clock and data to stabilize. To turn off the power domain, the reverse sequence could be applied. That is: the third SIB closes and freezes the register's data; the second SIB closes and freezes the clock; and finally the first SIB closes and de-asserts the power buffer. This provides graceful power shutdown. Note that at least the first SIB must always be in the active power domain so that it can turn on or off its associated power domain. Other network bits may indicate that the power domain should be active or inactive. This will allow the closing of an access SIB to enable instrument concurrence to not automatically result in the power domain being shut off.

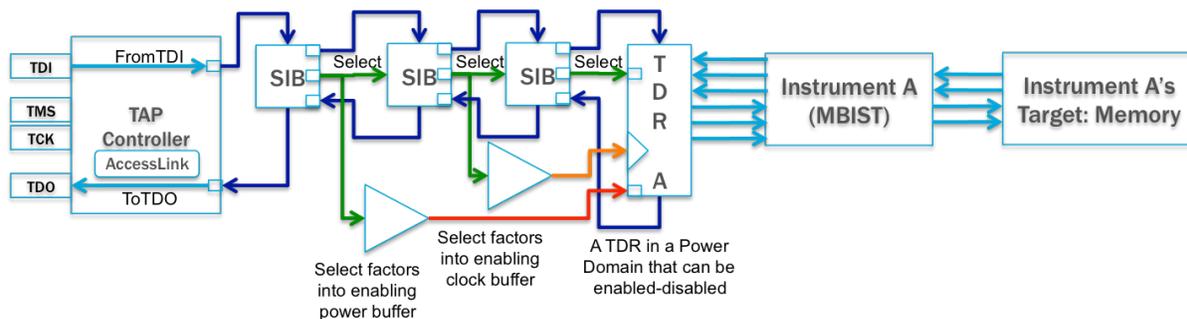


Figure 7: Implementing three SIBs to operate a register in a switchable power domain.

Note that a switchable power domain can be deployed in software. PDL instructions can write to scan path control bits that are not SIBs and the sequence of writes would emulate this hardware circuit. (To turn power on, first enable power, then enable clock and lastly enable the target TDR. To turn power off, the reverse is followed. If a reset occurs, which normally closes all SIBs, then power SIBs can factor their Select signals into the previous hierarchical SIB update cells so that the SIBs will close on reset with a time constant to maintain graceful shutdown.)

In summary, opening and closing SIBs on the active scan path allows the following:

- Direct management of the instrument network (sometimes referred to as the length of the scan chain, which also directly relates to the access time for any given instrument interface)

- Low-power clock and power domains may remain off while the rest of the network or scan chain operates
- Instruments may continue to operate while the rest of the network or scan path is being used

Physical Tradeoffs

In addition to scan length management, power domain segmentation and instrument operational concurrence, a SIB also generates a local IJTAG Select signal so that an instruction decode that would normally be performed centrally in the device's IEEE 1149.1 JTAG TAP Controller will now be performed by the SIB's Select signal that is physically local to the instrument and near the TDR to which the SIB is providing the Select signal. (Figure 8). This is possible as long as JTAG's ShiftEn, CaptureEn, and UpdateEn signals are routed as generic clock-like signals. Generating a Select signal in a SIB that is close in proximity to the target TDR positively affects the chip design's physical and engineering tradeoffs, such as silicon area, wire routing, routing congestion, signal timing and power consumption.

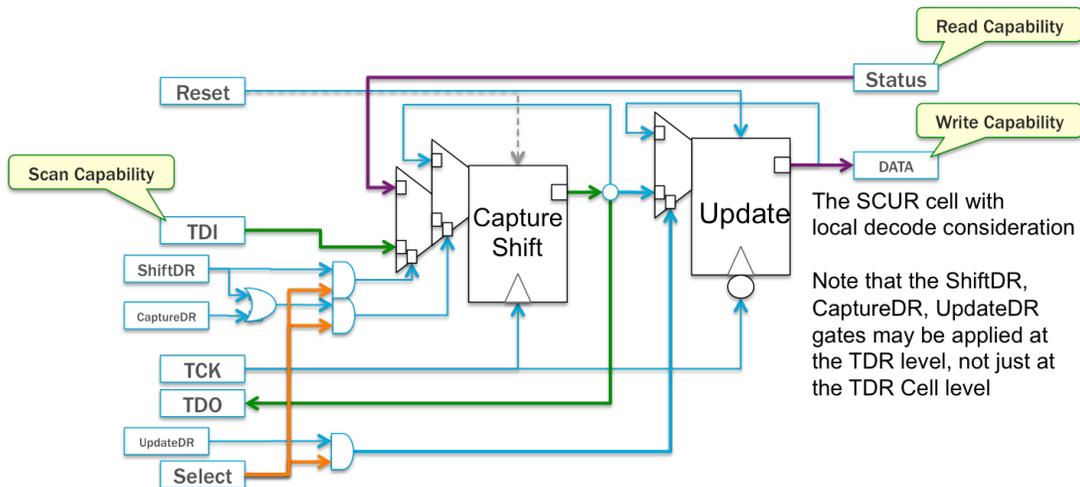


Figure 8: A Shift-Capture-Update-Reset TDR instrument interface cell with local decode.

One of the powerful aspects of IEEE 1687 IJTAG is its ability to reduce route congestion on a chip. The IEEE 1687 Select signal accomplishes this optimization because it enables the ShiftEn, CaptureEn, and UpdateEn operation signals to be gated or enabled locally to the TDR that interfaces to an embedded instrument. This differs from JTAG's methodology where these signals are gated and enabled within the centralized JTAG TAP controller block. Normally, each

instruction within an IEEE 1149.1 JTAG TAP controller's IR decodes a unique set of ShiftEn, CaptureEn, and UpdateEn signals that come from the TAP controller's FSM. For example, if 200 instructions are needed, then 200 sets of signals are decoded near the TAP's IR and FSM and are distributed from the TAP controller through a wire bundle that includes unique decoded CaptureEn, ShiftEn, UpdateEn, TCK, TDI, TDO, and Reset signals at the very least. This is one of the reasons why the IEEE 1687 IJTAG working group judged IEEE 1149.1 JTAG to be non-scalable and insufficient to meet the growing needs of embedded instrumentation.

The strategy implemented in the IEEE 1687 IJTAG architecture was that one instruction, the AccessLink instruction, would be required instead of the multiple instructions (like the 200 or more in the example cited above). As a result, the ShiftEn, CaptureEn, UpdateEn, and Reset signals would be distributed directly from the TAP's FSM without decoding in the TAP controller. This would require routing as an optimized, timing-managed tree structure, which could also reduce power consumption and could possibly increase the operational frequency for TCK on the IJTAG network. In an IJTAG network then, the Select signal can be generated within an embedded instruction bit that is part of the data scan path. For example, a SIB could open a segment of the scan path. The newly opened segment can then have its operation signals, the ShiftEn, UpdateEn, CaptureEn and Reset signals, which are synchronized with TCK, gated locally to the TDR (Figure 9). By moving decode out into the chip and away from the centralized TAP IR and FSM, the chip design is relieved of potential routing congestion problems.

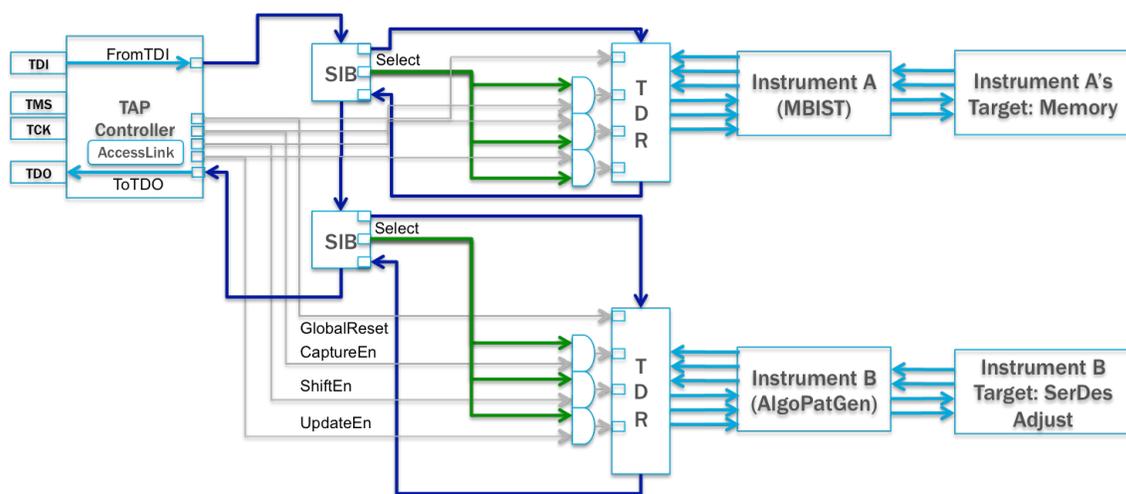


Figure 9: Example of locally generated Select signal from a SIB and moving decode closer to an instrument's TDR.

The IJTAG standard's concept of a SIB provides a wealth of opportunities for optimization, but including these SIB-related capabilities meant that the standard needed the ICL language to describe variable-length scan paths and to give engineers the ability to nest variable-length scan paths. The result was a realizable architecture and a language that could describe a hierarchical IJTAG architecture or network of scan paths where the various levels in the architecture are accessed by SIBs. For example, one SIB might open and provide access to additional SIBs and behind these SIBs there might be another series of SIBs. To accomplish this, the IJTAG working group had to violate a basic rule of the IEEE 1149.1 JTAG standard, which requires that instructions and data should be treated separately. IJTAG departs from JTAG in this regard because a SIB is actually an in-line embedded scan path instruction bit that configures the scan path during a JTAG data scan (DR-Scan) and not an instruction scan (IR-Scan). ICL was the first step within the IJTAG standard toward instrument and network re-use.

Embedded TAPs

When the IEEE 1149.1 JTAG standard was defined in the 1980s and early 1990s, it limited the number of TAPs per chip to one. An embedded TAP (eTAP) is defined as an IEEE 1149.1 JTAG TAP that is not the main TAP and TAP controller on the chip and which is integrated internally into the chip design (Figure 10). A multi-TAP architecture is defined as a chip design with more than one embedded IEEE 1149.1 JTAG TAP in addition to the main TAP and TAP controller on the chip. These eTAPs are integrated internally into a chip design. This has taken on critical importance as many of the embedded instruments and other cores that are provided as IP today include an IEEE 1149.1 TAP so that the IP can be accessed via JTAG. When multiple instruments are embedded into a chip design and each instrument IP has its own JTAG TAP, the rules of the IEEE 1149.1 JTAG standard are violated. (Note that the JTAG standard allows for 'compliance enable pins,' but adding pins is expensive and not scalable.) The IJTAG standard clarifies this situation. A device's primary JTAG TAP provides access to the entire device while all eTAPs within the device are subservient to the primary TAP.

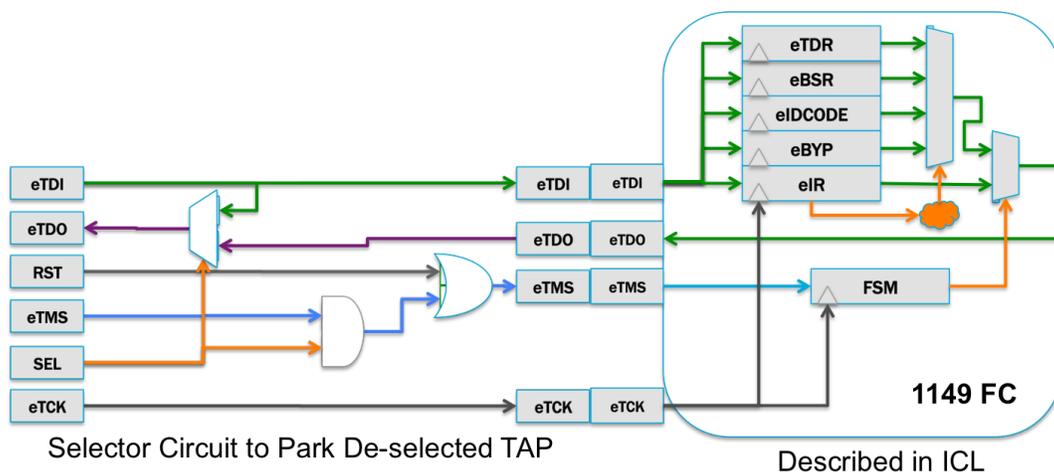


Figure 10: An embedded fully-compliant IEEE 1149.1 JTAG TAP and TAP controller.

The rules pertaining to eTAPs in the IJTAG standard parrot the standard's rules for SIBs. For example, when an eTAP is deselected, it is effectively frozen (meaning it processes no Shift, Capture, or Update operations) and it is no longer part of the active scan path. When awakened, an eTAP synchronizes with the primary TAP. Deselecting and selecting an eTAP is most effectively accomplished when a deselected eTAP is parked in the RTI state and the chip's primary TAP is required to leave Update-DR/IR and proceed only to the RTI state. If the embedded TAP does not support the TRST* input signal, then an additional requirement is placed on the main TAP. Namely, it must generate an embedded TAP reset by entering the TLR state and must remain there for at least five TCK cycles to ensure that embedded TAP controller state machines also enter the TLR state. Alternatively, when the main TAP's TRST* signal is asserted (if supported), it must similarly provide a reset signal for at least five TCK cycles so the eTAPs can enter the TLR state as well.

To ensure that an eTAP will function properly, the selector circuit shown in Figure 10 should be included in the design. This signal mirrors the chip-level JTAG TAP interface with the addition of the Select signal (shown as SEL). Similarly to a SIB, this Select signal can add the eTAP into the scan path when asserted and remove it from the scan path when de-asserted. This Select signal also places the TMS in a logic 0 state, which keeps the eTAP parked in the RTI state when de-asserted. Note that a reset operation or signal from the main TAP controller (shown as RST) overrides the embedded TMS (eTMS) and can place the eTAP into the TLR state if the main chip TAP issues a reset by entering TLR itself or, if supported, by asserting its TRST* signal.

This notion of an on-chip architecture with multiple TAPs resembles a scaled-down and internalized version of a printed circuit board design with multiple JTAG scan path linkers. Approached from this perspective, ICL could describe several straightforward architectures with various hierarchical configurations of eTAPs. To facilitate multiple eTAPs, an embedded TMS (eTMS) signal was added to the IJTAG standard's plug-and-play network interface. (TMS refers to JTAG's Test Mode Select signal.) Note that in Figure 11 all of the normal TAP signals are preceded by an 'e' to indicate embedded and to label these signals as embedded TAP signals. For example, the signals from an eTAP are represented as eTDI, eTDO, eTMS, and eTCK.

eTAPs are another example of how the IJTAG standard ensures the portability of instruments and instrumentation networks. An eTAP may represent a fully compliant JTAG architecture or it may contain IJTAG AccessLink instructions to make it a JTAG/IJTAG architecture. Figure 10 shows how the main TAP on a chip can have an added Configuration Register that is in series with multiple eTAPs in a daisy-chain configuration. This allows the eTAPs to be enabled and disabled by the contents of the main TAP's Configuration Register. When the main TAP's FSM is on the IR-side, then all of the IR's for the active TAPs are in series with the main TAP's IR. When any instruction is active, the selected eTAPs are in series with the selected TDR register. Special accommodations should be made to allow the Configuration Register to be 'zeroed-out' if certain main TAP instructions should not access the eTAPs. For example, the BYPASS instruction should zero-out the Configuration Register to ensure that only one bypass register is seen at the board level. Further exploration on the topic of eTAPs can be found in another eBook in ASSET InterTech's extensive eResources section on its website. Here is a link to "[A Modest Proposal for Chip-Level Consolidation of a Multiplicity of Embedded TAPs with consideration for JTAG Boundary Scan, JTAG Software Debug and IJTAG Instrumentation Networks.](#)"

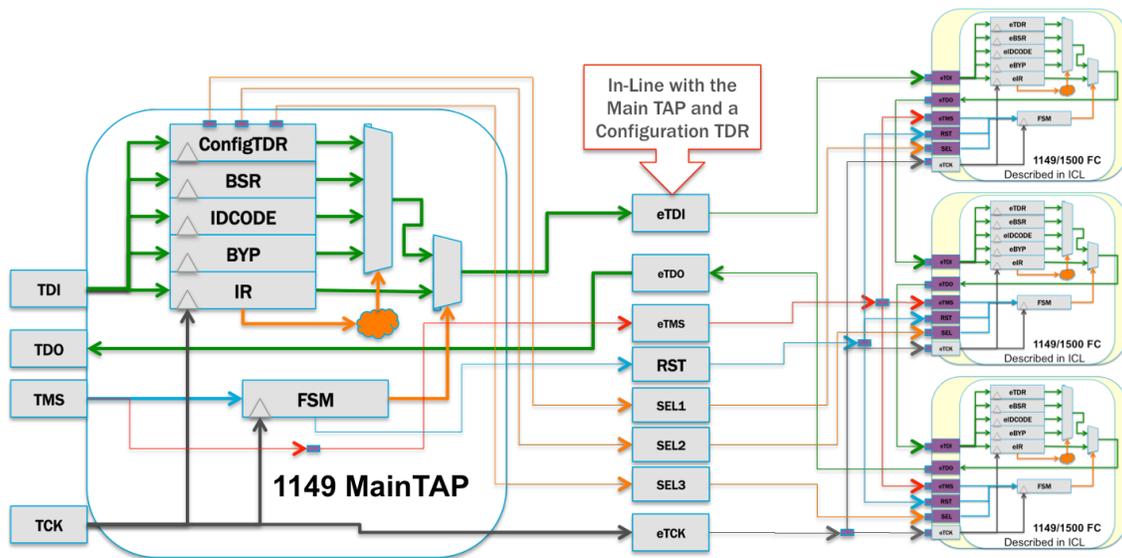


Figure 11: A main TAP accessing three eTAPs in a daisy-chain connection.

Simple and Advanced Concurrent Instrument Operations

Simple concurrent instrument operations has been mentioned previously in this eBook. This occurs when multiple instruments are accessible during a DR-scan so that each instrument's PDL iWrites and iReads can be scheduled for application simultaneously. Multiple instruments can be operated at the same time by initiating each instrument with the same DR-scan. Alternatively, their operations can be scheduled to begin at different times or with overlapping operations with different DR-scans.

More advanced concurrent instrument operations occur when instruments operate on conditions relative to each other. For example, a requirement or the availability of a resource could prevent instruments from operating concurrently. Consider the availability of a phase-locked loop (PLL) as a requirement. If one instrument requires the chip's PLL to be set to operate at 125 MHz and to generate a 125 MHz clock signal while another instrument requires that the same PLL operate at 400 MHz, then these two instruments cannot operate concurrently. (Of course, one instrument could operate at the wrong clock setting, but its operation would certainly not meet expectations.) This type of instrument operation is called 'restrictive' concurrent operation.

To a certain degree, some rudimentary restrictive concurrent operations are enabled by the IEEE 1687 IJTAG standard. This rudimentary handling of concurrent operations uses the advanced

PDL commands of iTake and iRelease. In a basic sense, an iTake command is like reserving a hotel room or a seat on a plane. The room or seat is reserved until it is released by the person who made the reservation. The iTake command reserves a shared resource such as a PLL for the requesting instrument. If an operating instrument has issued an iTake, then the other instruments on the chip will not be able to schedule the resource until an iRelease is issued by the initiating instrument. An iRelease makes the resource available to other instruments. If a test program is written in such a way that it attempts to dedicate a restricted resource to conflicting instruments simultaneously, the solution will be that the instrument that does not have control of the restricted resource through an iTake will be prevented from issuing iReads and iWrites by the IJTAG retargeting software that builds the DR-scans to fill the bits in the active scan path. The iTake and iMerge command are meant to convey this information to the retargeting software so that the real-time building of DR-scans to be applied to the IEEE 1687 IJTAG network will not actually allow conflicting instruments to operate concurrently.

Of course, much more would be involved in complex concurrent operations, but at least some level of synchronized concurrent operations for embedded instruments and for eTAPs has been enabled in the IJTAG standard. The issue of synchronized concurrent operation is larger than what was addressed in the initial version of the IJTAG standard, but the ground rules have been established. A limited amount of concurrent operation handling can be accomplished with IJTAG's PDL commands of iTake and iRelease and their resource allocation schemes.

What's next for the IJTAG standard?

Any discussion of topics such as multiple eTAPS embedded on the same chip and synchronous instrument concurrence inevitably leads to questions of how the IEEE 1687 IJTAG standard will evolve in the future. Following their official publication dates, most IEEE standards are expanded with new capabilities or features in subsequent versions. Often, the working group that developed the initial version of a standard likely identified many of the standard's subsequent new features during development of the initial version, but postponed the work needed to include them so that the initial version could be published in a timely manner. In addition, working groups frequently take into consideration feedback from the first users or early adopters of a new standard who usually discover where and how the standard might be improved. No matter where

the suggestions for improvements come from, the point is that standards evolve over time to keep pace with the progress of technology in the industry. And this is where the IEEE 1687 IJTAG standard is today.

The IJTAG standard was developed to manage and operate instruments embedded within chips. Many of these instruments were developed and inserted only to help characterize, test and validate the host device. These embedded instruments were targeted for use on automatic test equipment (ATE). Little or no thought was given to how these same embedded instruments might be deployed beyond the chip level. For example, they might be applied to verify high-speed interfaces such as SATA or PCIe on system boards.

However, some embedded instruments and their procedures could be re-used again after chip test when the IC has been placed on a circuit board. At that point, the embedded instruments may be employed in board validation, test and debug applications, as well as to debug system-level software and firmware. This is where the value of the IJTAG standard can be extended by expanding it in the future so that various board-level technologies are

Why don't all chips have a JTAG TAP?

The IEEE 1149.1 JTAG standard requires that the entire architecture must be deployed, including the four-pin TAP (TDI, TDO, TMS, TCK) and the standard's register structure, consisting of an instruction register, bypass register, IDCode register, and even the boundary scan register. Some ICs are small or inexpensive and cannot afford four pins for JTAG; other ICs either do not support digital pins or they support only a few digital pins, so they do not support a boundary scan register. For example, digital-to-analog converters and analog-to-digital converters are largely considered analog parts and so have no or relatively few digital pins. Despite this, these types of devices may still have internal test and configuration functions.

able to access on-chip IJTAG resources and apply them to validation, debug and test processes outside of the chips where the instruments reside. Currently, the access mechanism taught by the initial version of the IJTAG standard is only the IEEE 1149.1 JTAG TAP, but the IJTAG standard was developed so that other access methods might be accommodated in the future even though they are not described in the current version of the standard. For example, those chips that do not include an IEEE 1149.1 JTAG TAP (See "Why don't all chips have a JTAG TAP?") might feature another communication interface such as SPI or I2C. In addition, some chips might not have a dedicated test port at all. Many devices like DRAM memories only feature a

functional interface and nothing more. The chips without a JTAG TAP port may still include embedded instruments which could be applied at the board level, and which could benefit from the portability and documentation associated with the IEEE 1687 IJTAG standard.

If the chip design team intends that the embedded instruments within a certain IC will function in isolation or independently from all other instruments inside the host IC or within other chips on the same circuit board, then concurrent instrument synchronization is not needed. But if instruments need to be started at the same time, for example, to supply a higher or more realistic activity level while instruments are operating; or to actually interact with each other through data transfers, then this type of functionality can currently only be performed under the IJTAG standard in a limited way and only when the instruments are all accessible through an IEEE 1149.1 JTAG TAP. For example, if some chips on a circuit board have JTAG TAP interfaces, while others have SPI or I2C ports, concurrent embedded instrument synchronization will probably be very difficult, especially when multiple hardware sources are driving the different types of ports on the chips on the board.

Synchronous Instrument Concurrency

Synchronous instrument concurrency is the ability for the embedded instruments within a chip or for embedded instruments within the chips on a circuit board to coordinate their activities, share data and collaborate on both chip-level and board-level processes. Synchronous instrument concurrency is a complex task. Because of this, an easy plug-and-play solution was not fully implemented in the first version of the IEEE 1687 IJTAG standard. The first step toward synchronous instrument concurrency would be support for other access mechanisms in addition to IEEE 1149.1 JTAG.

So, for example, interfacing an IJTAG network, which is currently based on the JTAG control protocol (capture, shift, update, reset), to a SPI slave interface is not straightforward at this time. It must be noted that a SPI slave is a selectable interface that would look very much like a selectable IJTAG eTAP. In this vein, there may be data staging, data packets and different sequencing involved with operating the network and delivering data to or extracting data from embedded instruments. Ultimately, a method may be needed to synchronize IJTAG iReads and

iWrites behind a SPI or I2C interface with iReads and iWrites behind the IEEE 1149.1 JTAG TAP controller. At the very least, a conversion FSM is needed that can take SPI data, clocking, and selection from the four-pin SPI interface consisting of MOSI, MISO, SSEN, SCLK signals and turn them into JTAG's CaptureEn, ShiftEn, UpdateEn, and Reset operation signals. Also, the FSM would need to pass data into the TDI and extract data from the TDO network signals, while supplying a TCK. When this FSM is available as well as ICL description of the internal access network and PDL description of the instrument's operations, then the relationship between the port and instrument issuing the iReads and iWrites can be established.

Future IJTAG Standards Activities

As the IJTAG standard currently stands, coordinating the activities of the instruments embedded in chips with different access mechanisms is beyond the scope of the standard, but this is a capability that well might be included in a new updated or incremental version of the IEEE 1687 IJTAG standard, such as IEEE P1687.1, for example.

One promising pursuit for expanding the value of IJTAG would be a study group that would define the rules, options, allowances and permissions involved with supporting alternative controller interfaces in an IJTAG on-chip network of embedded instruments. Any chip may have instruments and any chip may benefit from an on-chip management network, even if that chip does not support an IEEE 1149.1 JTAG TAP and TAP Controller. Such a study group for the next version or extension of the IEEE 1687 IJTAG standard would need to define the following:

1. A precise method for handling IJTAG's AccessLink for alternative serial controllers such as SPI or I2C. Alternatively, if there is not an instruction that would represent an AccessLink, then a new description element in IJTAG would be needed to describe how a SPI, I2C or other interface would provide the selection or access to any specified embedded instrument.
2. The conversion rules needed for alternative serial interfaces to directly drive an IJTAG network. These rules would define the relationship between the source protocol such as SPI, for example, and a target protocol like the IEEE 1687 IJTAG Network Capture-Shift-Update-Reset operations. The rules may be flexible to the point that they function as a description of the conversion state machine.
3. The rules or allowances for subordinating or nesting a JTAG eTAP controller under an alternative serial controller like SPI or I2C.

4. The rules or allowances for subordinating alternative controllers within an IEEE 1687 JTAG network in a manner similar to an embedded TAP (that is, creating an eSPI or eI2C, for example).
5. Hardware extensions to allow any other scan path or network modification elements required to implement or enable alternative access to embedded instruments. These might be staging elements and bidirectional control logic for dealing with bidirectional single-wire ScanIn and ScanOut pin ports, for example.
6. Extensions to ICL for describing the specific structures for including alternative controllers. These extensions would enable descriptions of ports, pins, pin directions, pin functions like Data, Clock, Select, Enable, Direction and others, as well as pin-allowable values such as 1/0/Z.
7. Extensions to ICL for describing protocol or conversion state machines other than the IEEE 1149.1 JTAG FSM. For example, ICL might describe how to convert SPI data or data packets into CaptureEn, ShiftEn, UpdateEn, and Reset operation signals.
8. Extensions to PDL, if they are needed, to accommodate alternative controllers. This might involve new keywords to deal with asserting or de-asserting bidirectional control.
9. Extensions to PDL to enhance synchronized concurrent operation among embedded instruments within the same IC or in separate chips on the same board. This might involve keywords or procedures for staging iWrites so that embedded instruments under different controllers will start on the same clock cycle. In a similar way, iReads should be completed at the same time. This would enable communication among embedded instruments through semaphore signals so that iProcs will return a value.

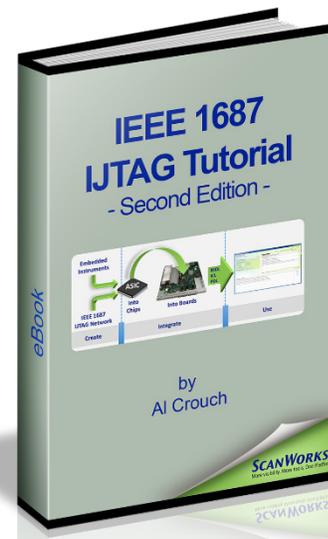
Conclusions

The ratification of a new standard is the beginning of a journey. Typically, a working group developing a new standard must draw the line somewhere in order to complete a first version. Those features that were not included in the approved version are left for subsequent additions to the specification. And, once engineers begin using the standard, other features and capabilities, as well as refinements to the features already included in the document, are soon identified. As a result, the subsequent revisions to a standard can often be as useful as its initial version. For example, consider the IEEE 1149.1 JTAG standard. Following its approval in the late 1980s, periodic revisions and extensions have been made until the most recent version was published in 2014. In all likelihood, IJTAG will follow a similar course because the importance of embedded instrumentations is only going to increase in the years ahead.

Individuals who are interested in joining or following a study group for IEEE 1687.1, tentatively named Alternate Serial Access for Embedded Instruments, should follow this [link](#) and fill out the registration form. (The information provided will remain private and only available to the IEEE IJTAG Study Group. It will not be used for marketing or other purposes.)

Learn More

For more information on the IEEE 1687 IJTAG standard, download the only Tutorial based on the approved version of the standard.



[Register Today!](#)