Statement of Work (SOW)
Test TIG
Built-In Self-Test (BIST) Project, Phase 3
Short-Term and Long-Term Strategies for
Use Case Standardization

Version 2.0
Date: July 17, 2012
Project Leader: Zoë Conroy, Cisco
Co-Project Leader: Alfred L. Crouch, ASSET InterTech, Inc.
iNEMI Staff: David Godlewski

Basic Project Information

Background/Context
This program's focus is on “Chip” Built-In Self-Test (BIST) study and its promotion for board and system-level applications.

Presently, there are no Standard Chip level interfaces or algorithms for BIST, which limits the introduction of BIST at board level test. Most “chip” level BISTs are designed to aid IC manufacturing; these algorithms are often not suitable or available to run at the board level.

The goal of this program is to:

- Develop and promote the adoption of IC BIST at the board/system level
- Encourage IC vendors to provide standard chip BIST interfaces and algorithms
- Encourage ATE/Instrument providers to develop products based on existing related standards for BIST design. For example, an IEEE 1500/P1687 globalized Test Cost Model useful throughout the industry

Purpose of Project

Presently, there are no Standard Chip level interfaces or algorithms, for Built-In Self-Test (BIST), which limits the introduction of BIST at board level test. Most “chip” level BIST systems are designed to aid IC manufacturing; these algorithms are often not suitable to the PCB board, or are not available to run at the board level.

The goals for this phase are to continue the iNEMI project to

- Develop and promote the adoption of chip BIST at the board/system level,
- Encourage IC vendors to provide standard chip BIST interfaces and algorithms,
- Encourage ATE/Instrument providers to develop products based on existing related standards for BIST design, for example, an IEEE 1500/P1687a globalized Test Cost Model useful throughout the industry.
- This group will also present to the industry relevant “use-case” models that will provide standardization to drive any needed changes to existing industry standards.

**Project Areas / Priorities**

iNEMI BIST Program consists of four phases:

1. **Survey on BIST availability, usage, access at board level test (Phase 1 - complete)**
2. **BIST Use Case Investigation Project (Phase 2 - complete)**
   Investigate and identify a “use case” to use as discussion, modeling, and development material. Also identify main users of board BIST, definition for board BIST, and its value add.
3. **BIST Use Case Function Classification Project (Phase 3 - this project)**
   Investigate and identify for the “use case” as defined in the iNEMI BIST Investigation Project (Phase 2) the tasks, tests, logic and features required and the access, control and vectors available.
4. **Board Level Test Recommendations for Standardization of Component BIST (Phase 4)**
   For the “use-cases” defined in Phase 3 (Phase 4) define standardized board level language. Feasibility study including IEEE 1687 IJTAG, IEEE Standard Test Access Port Boundary-Scan Architecture 1149.x, and IEEE 1500 Standard for Embedded Core Test (SECT).

Appendix A has details on BIST background and definitions.

**Scope of Work – Phase 3**

**Introduction**

Phase 2 results concluded; (1) the ability of BIST to test the interfaces between two adjacent ASICs and (2) from an ASIC to external memory interfaces.

These two use-cases will be the focus for Phase 3 work.

The main users of BIST are engineers in the NPI and board test environments; value-adds stated by these engineers were not only test time and cost reductions, but also the ability to add increased test coverage and diagnosis to existing test methodologies.

The definition of board level BIST or BA-BIST (Board-Assistance BIST) agreed on in Phase 2, and used here in the Phase 3 SOW, is as follows: **Board-Assistance BIST (Built-In Self-Test) is an embedded capability within an IC that is fully or partially self-contained, in that it incorporates some or all of the following capabilities:**

- Pattern/signal generation
- Pattern/signal delivery
- Response or data capture
- Response evaluation functions

Phase 3 will focus on a two tier strategy for both the short term and long term for BA-BIST usage at board and system level test. The short term and long term will be worked on simultaneously, as much as possible. However, the short-term work will continually feed into the long-term scope.

**NOTE: All changes to SOW must be approved by the Technical Committee for version control**

Built-In Self-Test (BIST) Project, Phase 3  Page 2 of 15
Short-Term Strategy: Background

The short-term strategy is directed toward the use of existing BIST available in devices today, which is in response to the results of the first two surveys.

For supplier-specific BIST, the tool investment is application specific with little ROI. The encouraging point is that the tools do exist. They are internally developed tools or APIs (Application Programming Interfaces) typically tasked to access the BIST for design, debug and validation purposes. These tools are often not ready for external use and some development effort may be required to make them user friendly, protect any IP (Intellectual Property) and provide documentation. If the board test process requires the use of BIST, in the short term, a request would be made to the device supplier to supply the tools to use that BIST. It is currently up to the end user of the BIST to make these requests of the IC supplier for tools and documentation to use the BIST effectively.

Short-Term Strategy: Statement of Work

While the team received good information from the two surveys on BA-BIST access and the most needed tests for board and system test, the iNEMI team still requires more in-depth information on the test features and functions, priorities, security requirements, etc. Further PR (public relations) on the BA-BIST value-add is also needed across the industry as a whole. With this in mind, the short-term strategy will address both of these.

Task 1: Project team plans to reach out to top IC suppliers to communicate the value add of BA-BIST to solve board test problems.

a. Potential Supplier list is Intel, TI, Freescale, On-Semi, Broadcom, Qualcomm, Marvel, STM, IBM, ARM, PLX, Cavium, Altera, Xilinx, Agere, AMD, Cypress, Fujitsu, Fairchild, Infineon, Lattice, Linear Tech, Nat Semi, Samsung, LSI, and Avago. Tool companies.

b. Supplier list to be categorized according to being BIST friendly, close partner with a team member, technology segments, to work with “easy to convince” suppliers first.

Task 2: Team to develop a standard presentation/talking points for Suppliers. Goal of presentation content:

- Buy in on BA-BIST value-add.
- Specific feedback on use-case features and standardizations.
- Bring out any dependencies per market type (e.g., cell phone vs. networking supplier).
- Obtain Feedback on whether Suppliers have public or private process to implement BIST and if they are comfortable with sharing it.
- Possible discussions on value-add of P1687 as part of a solution.
- More specific data for long-term strategy use case standardization.
- Address “fears” of supplier.

Task 3: Team to do further PR with industry on BA-BIST value. Any of the following will be chosen:

- Team to present whitepaper at ITC (Nov 2012) to further outreach to design and test community value of BA-BIST, and be a requirements doc for what’s needed to run BIST at the board level.
- Team to request an ITC (International Test Conference) panel.
- iNEMI Forum at ITC will have BIST section.
- Submit abstract to IEMT Conference (Nov 2012).
Whitepaper for popular Industry publications (IEEE Spectrum, Test and Measurement)
VTS 2013 paper and panel.

**Task 4:** Develop Position Paper by team based on surveys and short-term findings.

### Long-Term Strategy: Background

The long-term strategy is to develop tasks and tests based on the BIST use cases obtained from the survey in Phase 2. The use case solutions will address the major industry issues identified also from the surveys, with details on what kinds of instruments and IP solves them (e.g., the memory BIST use-case will craft and specify what is needed for a “standardized” board test).

These use-case standardization solutions will need to be socialized with end customers (e.g., Freescale, LSI, etc.) to get feedback on how difficult the function is to include, is it proprietary, can it be supported and other issues. Some of this feedback will come out of the short-term SOW above, hence the need to run the short term and long term in parallel. The team will run a test case of the use case on an FPGA board, to verify it does the right thing.

### Long-Term Strategy: Statement of Work

The output of the short term will continually feed into the long-term SOW to keep shaping the long-term work with supplier and industry feedback to provide the optimum solution. Hence, the tasks below will be expanded at the end of the short-term phase in March 2013. The thrust of the work will be to investigate and identify as defined in the iNEMI BIST “use case” Investigation Project (Phase 2), a function classification of the “use case” that includes the following tasks:

**Task 1:** Tasks/tests (i.e., types of BIST functions)
- List of tests and tasks performed for each use case. Including resets, pattern types, initializations, boot sequences. Tasks and tests to consider use with emerging technologies such as 2.5D/interposers, 3D, Photonics, etc.

**Task 2:** Logic/features involved in the tasks and tests
- List of logic and features needed for each test. For example, pattern generation, loopbacks, pattern compares, test, function and feature access, security of IP and path to IP.
- List what functions exist today, what is on horizon (i.e. roadmap). What exists today does not match all board test needs (e.g., memory BIST test patterns cannot diagnose pin issues) that are needed at board test. Recommendations for modifications to tests, functions and pattern to line up to board test needs.
- Logic and features must take into consideration use with emerging technologies such as 2.5D/interposers, 3D, Photonics, etc.

**Task 3:** Access, control and configuration requirements
- Test, control and configuration requirements

**Task 4:** Descriptions and Vectors that are available
- Language to describe the test and intent of the test. Lower level information required to describe the operation of the test features and information concerning the configuration.
- Documentation should cover functions not usually documented with a chip for standalone chip use, but are required for board level test use (e.g., ASIC to memory interface may contain external MBIST in the ASIC to test the external memory).
Task 5: Run test case on FPGA board

Task 6: Prepare report on use-case standardization requirements for appropriate standards body

These tasks should result in the identification of what can be standardized (board level language for chip to chip transactions).

**PROJECT IS / IS NOT Analysis**

<table>
<thead>
<tr>
<th>This Project IS:</th>
<th>This Project IS NOT:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-In Self-Test (BIST) Project, Phase 3:</strong></td>
<td></td>
</tr>
<tr>
<td>Develop guidelines and recommendations for new standard for PCBAs to run BIST tests based on component BIST use-cases defined in Phase 2.</td>
<td>Not a standards development</td>
</tr>
<tr>
<td>Transfer of the guideline and recommendations to the appropriate standards body for implementation.</td>
<td></td>
</tr>
<tr>
<td>Outreach to global electronic DfX (DfT and DfM, repair) industry on value add of BA-BIST.</td>
<td></td>
</tr>
<tr>
<td>Resolution of iNEMI 2009 Roadmap Gap Analysis.</td>
<td></td>
</tr>
<tr>
<td>Treating silicon as black boxes.</td>
<td></td>
</tr>
<tr>
<td>Strawman Guidelines on security aspect on implementing BA-BIST.</td>
<td></td>
</tr>
</tbody>
</table>

**Business Impact**

BIST’s use model is complicated by the fact that this is supplier-specific. Tool vendors do not build tools around supplier-specific BIST, as they would with industry-standard BIST. The investment made for a standardized BIST has tremendous ROI (Return on Investment) opportunity, across many devices, development phases and application segments.

**Participating organizations**

iNEMI member companies –

- Agilent Technologies
- Alcatel-Lucent
- ASSET InterTech
- Cisco Systems
- Corelis
- Dell, Inc.
- Delphi Electronics & Safety

Flextronics International
- Hewlett-Packard Company
- Intel Corporation
- Lenovo
- Quanta Computer
- Teradyne, Inc
Outcome of Project

Short Term

Papers for IEEE Conferences ITC 2012, IEMT 2012, iNEMI position paper to address industry requirements, other Industry Publication Papers (e.g., IEE Spectrum, component supplier outreach and communication).

New information collected from any of the above items on task/tests/features for the two use cases will drive updates and additions to the Long-Term SOW.

Long Term

Two BIST use cases (derived from Phase 2) with standardization of tasks/tests and features required for board test that will be used to drive new and existing recommendations for standards modifications.

The short-term output will help fully develop the long-term SOW and associated output.

Previous Related Work


– Cisco paper about ICT and BIST: http://www.molesystems.com/BTW/material/BTW07/Presentations/BTW07-Presentation%206.1.pdf

– Analog BIST: http://www.tmworld.com/blog/Taking_the_Measure/22566-Whatever_happened_to_analog_BIST_.php


– ITC paper from Sun/Microsoft about board/system level test of external memories: http://www.itcprogramdev.org/itc2003proc/Papers/PDFs/0037_2c.pdf


Prospective Participants

1. Current companies participating on the Project Formation Team are:

NOTE: All changes to SOW must be approved by the Technical Committee for version control
Built-In Self-Test (BIST) Project, Phase 3 Page 6 of 15
2. Participant companies should also include:
   - Chip design companies (memory, microprocessor, ASIC, datacom)
   - TI, Micron, Marvell, Broadcom, IBM, LSI, Maxim, Micrel, Vitesse (ASIC vendors/manufacturers)
   - Board and system design companies (PC, networking, test)
   - Dell, Delphi, Lenovo, Quanta, Oracle, etc.
   - EMS, CMs, ATE Suppliers, ICT Fixture suppliers
   - Celestica, Flextronics, Foxconn, Plexus, Sanmina-SCI, Teradyne, STATS ChipPAC, Jabil, Advantest, etc.
   - Boundary scan suppliers
   - JTAG, Intellitech

3. All participating companies must be iNEMI members.

**Project Plan**

**Schedule with Milestones**
- Project plan with identified tasks, intermediate check points, and end dates
- Q1 is expected to be in the Sept-Nov 2012 timeframe

<table>
<thead>
<tr>
<th>Phase 3 Short Term</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Communicate the value add of BA-BIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 2: Develop a standard presentation/talking points for Suppliers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 3: Further PR with industry on BA-BIST value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 4: Develop Position Paper</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase 3 Long Term</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Tasks/tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 2: Logic/features involved in the tasks and tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 3: Access, control and configuration requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 4: Descriptions and Vectors that are available</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 5: Run test case on FPGA board</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 6: Prepare report on use-case standardization requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Project monitoring plans**
- Planned teleconference schedule: Weekly conference calls on Wednesday afternoons at 1:00 p.m. (PDT/PST); 5:00 am/6:00 am China/Taiwan.

**NOTE:** All changes to SOW must be approved by the Technical Committee for version control

Built-In Self-Test (BIST) Project, Phase 3
• Meeting minutes and agenda provided through e-mail.
• Follow-up with individuals on an as-needed basis.
• Workshops and face-to-face meetings as appropriate.
• Progress reports will be issued as tasks are completed.
• A mid-point progress update will be made to the iNEMI Technical Committee at the end of Short-Term Strategy.
• Provide quarterly reports briefly indicating progress. This could be a short series of PowerPoint slides showing the work in progress.
• Review all project requirements with prospective participants before the project begins.
• Ensure open lines of communication among participants.
• Provide any project specific monitoring or communications plans (e.g., multiple project meetings to cover multiple regions – EMEA, Asia, Americas).
• Progress reports will be provided upon request for presentation at regularly scheduled iNEMI meetings (e.g., a short series of PowerPoint slides showing the work in progress at member council meetings).
• Track and document approximate number of people on the project per quarter (can be tracked through iNEMI’s WebEx account).

General and Administrative

Guidelines for this project and all other iNEMI Projects are documented at
Appendix A

Program Background and Definitions:

**BIST Background**

BIST, as applied to Semiconductor ICs, is not one single feature or methodology, but there are many items that may be called BIST. The term BIST is an acronym for Built-In Self-Test, and in the IC world, this usually means Memory Built-In Self-Test (MBIST), Logic Built-In Self-Test (LBIST) and High-Speed I/O Built-In Self-Test (HSIO-BIST).

The term BIST may also be applied to any built-in or self-contained hardware feature within a semiconductor IC — but not all features have value beyond just semiconductor test and verification.

The processor-based, code-based, or externally-applied features, although capable of conducting the same operations, are generally not referred to as BIST by IC test professionals.

For the iNEMI effort, the BIST that is of interest are those embedded or built-in semiconductor features that provide value to conducting test, debug, and characterization in the board environment. At this point in time, the main BIST-type items that have been deemed as useful are the:

- HSIO-BIST for characterizing the at-speed signal-integrity of the traces on the board.
- The MBIST for evaluating if memory arrays on a board (whether inside of a chip or standalone as a chip) are in working order, are configured properly, and are repaired if applicable.
- External Memory BIST that tests an external memory that interfaces to another chip, as well as the interface itself.
- The LBIST for determining if a failing board test is due to the board or is the fault of a specific chip.
- MBIST or Memory BIST is a Structural Test applied Functionally meaning that the functional operations of Read an Address, Write an Address, and Select an Address are used to conduct the test, but the specific sequence of Reads and Writes and Addresses that are applied are meant to exercise the structure of the memory to find defect and fault-based problems (a side effect is that the functionality of the memory is verified).
- Memory BIST is the inclusion of specific hardware to generate the Write Data; Read Expect; Address Sequencer; and Compare Logic within a Chip or Board Design.
- Common Memory BIST (also called MBIST) algorithms for SRAMs are March3, March7, March14, Address Uniqueness, Walking 1, Filling 1, etc. and these algorithms are meant to identify word-line shorts and opens, bit-line shorts and opens, address-line shorts and opens, column and row driver slow-to-rise and slow-to-fall problems — these results can be mapped onto logic and physical bitmaps of the memory to conduct diagnosis.
- An evolution of Memory BIST used today is Memory Built-In Self-Repair (also known as BISR) which is the building in of features to allow redundant Rows and Columns to be mapped into the memory to replace faulty Rows and Columns — there may just be the access and ability to map the redundant Rows and Columns into the memory, or there may be a built-in algorithmic unit that determines which rows or columns are bad and determines the nature and configuration of the fix.
**Users and Uses of BIST**

BIST has different techniques, methodologies, and uses when applied to different Board Environments. For example, for New Product Introduction, the main goal is to evaluate the decisions made in the building of the board (chip placement, route lengths, etc.) and this requires using BIST for characterization tests to collect data about various choices and configurations.

Board Test is concerned about time-efficiently creation or the use of a test that operates within a minimum-time providing the most coverage, and can be resolved into a simple pass-fail indication.

The debug, diagnosis, yield-analysis and field-return analysis is all about using BIST to help with root-cause analysis whether by using divide-and-conquer or by collecting data for statistical data-mining analysis.

**Value-Adds**

To explain the ROI and Value-Add further: currently, determining whether “the signal integrity of a route between any two chips on a board is within operating bounds” cannot be done by using ICT probes on the route, and even adding probe pads to this route may change its ability to operate (makes it susceptible to electrical noise).

Because of this (without BIST), the only way to assess the signal integrity of these pins is to actually operate a completed and working board and to monitor the BER of the receiver chip.

HSIO-BIST allows a pattern-generator on one chip to directly operate the LVDS or High-Speed signals; allows another chip to either evaluate the pattern or to provide a loopback connection to allow the source chip to receive its own sent signal and to evaluate the pattern.

This type of testing is possible even if the rest of the chip is non-bootable (so, HSIO-BIST adds capability to develop and conduct a test that is not possible with today’s methods and it enables testing even if the rest of the board is not complete and the chips are not bootable).

The effect of having HSIO-BIST is to provide a test and test method that is much more efficient and timely in test development than the existing methods.

Instead of writing code to boot up the whole board and then writing operational code to deliver the vectors and to observe the BER signal — a simple 1149.1 type ScanIR and ScanDR can be auto-generated and applied to the chip through a JTAG Tester even if the board is not complete. The auto-generation and operation of JTAG reduces the knowledge requirement of the user from having to understand the board and the chip and how to write operational code — to only having to understand how to use JTAG.

In addition, JTAG Testers are often small, low-cost hardware units as opposed to high-dollar ICT Hardware, IC Test ATE, and traditional Board Testers.

Other value adds of BIST include faster test time than traditional functional testing, and often improved test coverage and defect isolation.

**Tasks/Tests**

At the Board Level, there may exist, several chips per board that may carry built-in BIST functions that may provide value in Board Test and Board Operation environments. Mostly, these functions are the aforementioned LBIST, MBIST, and HSIO-BIST. From a Board Level description, the change in point-of-view is now from using the BIST function to test the chip in a standalone manner to using the BIST function or coupling the BIST function with another chip to provide a “value-add function” for the Board.
The HSIO-BIST can be used to couple two chips together, one chip as a source of patterns and the other chip as either a loopback connection or as a pattern evaluation to verify the signal integrity of the board traces between the two chips (outwardly, even though the pattern is generated as a structural test, the test looks like a functional test between two chips).

The Memory BIST within a chip is used in a debug sense, to take a suspected memory error out of the equation, when board-level data transfers are in error. For Standalone Memories, that have no inherent test functions, then a Memory BIST function can be programmed into an on-board FPGA and this hardware-driven MBIST can be used to provide on-board self-test to the on-board Standalone Memory. This provides the function, of an easy to use test for a memory that otherwise would require a fully functional operational board-level code to be developed.

The Logic BIST within a chip is used to take a suspected chip out of the equation, when board-level data transfers or board operations are in error.

The Logic BIST and the Memory BIST may also be operated just to generate digital noise (power consumption, voltage-rail loading, clock-generation, etc.) at the board-level. While other tests and operations in other chips (such as an HSIO-BIST between two other chips) are active.

**Logic/Features Involved**

*Determine interface types covered*

For the targeted use-case example, the type of interface used must be easily available as a core, must be programmable into an FPGA, and must be familiar throughout the NPI and BT community. For this reason, the most-likely HSIO protocols to be used as examples are the PCIe and DDRx units.

*Logic block features for testability*

For the HSIO Use-Case Example, then the expected block features for testability would be the existence within the chip of a digital pattern generator, a digital data comparator, an RX to TX loopback connection, and adjustment control of the PHY. For chips that have advanced DFT features, it is expected that there may be other features such as error injection into the pattern generator, a selection of patterns or pattern sequences that can be generated, and a wider selection of loopback connections that may allow the patterns to be looped-back at different depths in the HSIO/SerDes/LVDS hierarchy.

The testability included within the chip to make the HSIO testable may have differences in specific implementation, but can be classed in a generic sense. For IC Testing as a standalone device on an IC ATE, the testability (known as DFT logic to the IC community) is generally a Pattern Generator and a Data Comparator that is associated with the last digital stage of the implementation and a loopback connection from TX to RX associated with the analog stage of the design.

* Structural Tests are defined to be tests that are developed to exercise defect and fault models; as opposed to Functional Tests which are defined to be tests that are developed to verify operational behavior. Generally, structural tests are smaller and more efficient than functional tests – for example, to statically (not at-speed) test a 4-bit adder functionally requires adding every pair-combination of numbers from 0 to F (F-squared number of vectors), where a fault-based test may determine all potential gate-level stuck-at faults with just 2 or 3 applied vectors. The structural test in this case would verify the truth-tables of all gates and that all gates have complete and proper input and output connections. The functional test in this case would verify that the collection of gates and wires actually implement an adder.
More specifically, the pattern generator is a counter or linear-feedback shift-register (LFSR) to create the TX output — and the comparator is a registered XNOR that uses a delayed version of the pattern generator output to also conduct a comparison on the RX input. The loopback connection that connects the TX output to the RX input may be included on the die or may be specified to be implemented externally on the ATE Loadboard.

To expand the chip’s test capability to include board-level traces, then a loopback from the RX to the TX needs to be supported — this allows an external analog signal that is received from one chip to be looped around to emulate a transmitted signal from another chip.

Simply operating the Pattern Generator and Comparator with the loopback connected verifies that the Digital-to-Analog conversion works, that the PHY output transfers the analog version of that data to the chip pins correctly, that the analog signals return back to the RX pin and then get reconverted back to digital data that matches the outgoing digital pattern — this type of test just verifies the Analog-to-Digital, Digital-to-Analog, PHY and loopback connection.

To characterize the PHY, the BER signal needs to be monitored and the ability to insert errors into the data stream may be needed — and the ability to adjust the PHY parameters needs to be included.

**Access and control configuration requirements**

The Embedded Testability Features, also known as Embedded Instruments, can have many different access mechanisms — some may be custom per each chip design, and some that obey an industry standard.

For access to embedded instruments at the Board Test Environments, only those that have dedicated pins at the edge of the chip can be used (those access mechanisms that require borrowing functional pins are not effectively usable since those pins will be connected to their true board purpose and it is too costly to include “extra” on-board multiplexed access to allow an alternate mechanism to be designed). This means that access and use of embedded chip features can only be supported if the embedded feature is accessible through a dedicated test port such as the 1149.1 TAP Controller, that is in addition, for the Boundary Scan testing (verification of chip I/O connectivity).

Even though the 1149.1 TAP defines a 4-pin, with optional 5th pin, port to access and operate control logic and to gain access to serial scan registers — it has been co-opted by chip designers to do double duty for test since shortly after its ratification as an IEEE Standard.

The original form of “additional functionality” took the form of adding extra “private instructions” which select “private scan registers” to be connected between the TDI and TDO pins.

Some of the Capture Cells or Bits of the private scan register can be used to “read” from the embedded instruments when a Capture-DR operation followed by a Shift-DR operation is conducted by the TAP State-Machine; and some of the Update Cells or Bits of the private scan register can be used to “write” to the embedded instruments when a Shift-DR operation is used to introduce new data into the scan register and this is followed by an Update-DR operation to apply the new data.

Later optimizations were made to this process which resulted in the IEEE 1500 architecture and the IEEE P1687 architecture being defined to not only allow for the “reading” and “writing” operations, but to handle the documentation and vector requirements.

These architectures allow for “embedded instruction registers” to be included more locally to the embedded instrument or the embedded core and allow for scan paths to change length when an Update-DR operation is conducted.
These architecture changes largely allow multiple embedded instruments to be accessed and operated simultaneously (test scheduling flexibility) since a compliant 1149.1 architecture only supports one instruction at a time.

In addition, many of these embedded instruments have “local vectors” that include not only read (capture data from instrument signals) and write (apply data to instrument signals) operations, but also flow-control operations (for-next, if-then-else, do-while, etc.) — the 1500 and P1687 vector portions of the standards include and enable the retargeting of local vectors.

Documentation required

In order to describe tests and operations, keeping in mind that this test is at the board level and targeted for NPI and BT but that the testability features are at the chip level or even at an embedded core level within a chip, then the documentation is actually a mixed set of languages.

To describe the test, a language must exist at the “board” or “system” level to describe the test and the intent of the test.

For example, for New Product Introduction one of the tests that is wanted would be the Bit Error Rate test, and at the board level this would be described as “Chip A transmits a pattern sequence to Chip B for 25 million clock cycles while monitoring the BER signal of Chip B — an assertion of the BER signal will result in a fail.”

A related test would be: “Chip A transmits a pattern sequence to Chip B for 25 million clock cycles while monitoring the BER signal and while intentionally inserting errors into Chip A’s pattern sequence — an assertion of the BER signal within 5 cycles of any inserted error will result in a passing test.”

These descriptions pass the intent of the test on to Board Test Development and Board Test Engineer.

However, to complete the tasks required to actually develop a test — the lower level information concerning the operation of the HSIO-BIST Testability features and the information concerning the 1149.1 configuration required to access the HSIO-BIST must be provided.

This means that the 1687 ICL description and PDL vectors must be referenced; and further, the 1149.1 BSDL description and SVF vectors must be referenced. If a complete test HSIO test were described, then the code description and progression may be represented as follows (but the actions may be played from the bottom up):

- Board: Chip A Transmits alternating 5-A Pattern for 25 Million Clock Cycles
- Board: Chip B Receives the pattern from Chip A and returns the pattern through a Loopback connection
- Board: Chip A Receives the pattern returned from Chip BChip: (PDL a): Chip A’s Pattern Generator is Reset – then Started – and self-runs for 25 Million Clock Cycles
- Chip: (PDL b): Chip B’s Loopback connection is Enabled
- Chip: (PDL c): Chip A’s Response Evaluator (Comparator) is invoked to evaluate data from the RX pin for 25 Million Clocks
- Board: PDL c and PDL b are triggered simultaneously
- TAP: (SVF x): Chip A’s Instruction <L1001> is installed
- TAP: (SVF y): Chip B’s Instruction <111000> is installed
Board Level Test Recommendations for Standardization of Component BIST

Example - Board level language for chip to chip transactions

There are two views to standardization — to standardize Legacy (existing) architectures and actions; or to look to the future and create a standard that will allow growth. Looking to Legacy means that the logic and test features already exist in some chips — looking to the future means that there might need to be a time period for chips to adopt the standardized access architecture and embedded instrument features.

There is a need, in a standards way of thinking, to document the BIST — the goal is two-fold:

1. To document the test at the board-level to let an end user know what test is available and what it is supposed to do (and how to do it).

2. To document the components of the test (e.g., which chips are involved, which units within the chip are involved, how to access the individual units). Today, the “inside the chip” units may be covered by:
   i. 1149.1 BSDL (pins, instructions, registers) and SVF (ScanDRs through selected JTAG Registers)
   ii. 1500 CTL (cores, instructions, registers) and STIL (Vector Data associated with Cores)
   iii. 1687 ICL (instruments, instruments signals) and PDL (Procedures and Operations associated with Instruments)

Today, there is no defined language at the board/system level other than software languages (C, C++, Python, Tcl, Ruby, Perl, etc.). What is needed is a Language that defines board components as objects and states the “interaction” between them:

A. Chip A = Object-A; Chip B = Object-B

B. Object-A Transmits to Object-B Data-A that verifies the signal integrity of the Board Traces between them
C. Object-B Transmits to Object-A the acknowledgement that it received Data-A and that Data-A is correct

D. Data-A is defined to be 488-bit long packets in the ABC-Protocol

E. Then there is under this, a description of the Object-A and Object-B resources and requirements needed for the interaction of:
   5a. Object-A uses P1687 Instrument-D that is described in ICL File-E and PDL File-F
   5b. Object-B uses P1687 Instrument-G that is described in ICL File-H and PDL File-J
   5c. Object-A PDL routine is named Start-Trace-Verification
   5d. Object-B PDL routine is named Start-Receive-Data
   5e. PDL Start-Trace-Verification and Start-Receive-Data must be asserted simultaneously