iNEMI Statement of Work (SOW)
Test TIG
Built-In Self-Test (BIST) Program
BIST Use Case Investigation Project, Phase 1

Version 2.1
Date: January 20, 2011

Program Leader: Zoë Conroy (Cisco)
Co-Program Leader: Alfred L. Crouch (ASSET InterTech, Inc.)
iNEMI Staff: David Godlewski

Program Scope of Work
This project’s focus is on “Chip” Built-in Self-Test (BIST) study and its promotion for board and system-level applications.

Presently, from the board development point of view, there are no Standard Chip level interfaces or algorithms for BIST, which limit 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
- Steer the IC providers toward BIST functions that are helpful 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 (e.g., an IEEE 1500/P1687 globalized Test Cost Model useful throughout the industry).

Project Areas / Priorities
The iNEMI BIST Program consists of three phases; an output of each phase will define the next phase’s SOW.
1. **BIST Use Case Investigation Project, Phase 1**

   Investigate and identify a use case to use as discussion, modeling, and development material including:

   a. A definition of IC BIST to be used for Board/System purposes
   b. Users of BIST (NPI, BT, BD&Y, RMA)
   c. Identification of value-adds (what does BIST provide for the users specified)
   d. Identification of BIST functions under consideration (HSIO-BIST, MBIST, LBIST)

   NPI = New Product Introduction
   BT = Board Test (includes Characterization, Specification Test, and Manufacturing Test)
   MBT = Manufacturing Board Test (one specific subset of Board Test)
   BD&Y = Board Debug-Diagnosis & Yield
   RMA = Return Material Authorization

2. **BIST Use Case Function Classification Project, Phase 2**

   Investigate and identify (as defined in the BIST Use Case Investigation Project, Phase 1) a function classification of the use case including the following:

   a. Tasks/tests (i.e., types of BIST functions)
   b. Logic/features involved
   c. Access, control and configuration requirements
   d. Descriptions and vectors that are available

   Resulting in the identification of what can be standardized (board level language for chip to chip transactions)

3. **Board Level Test Recommendations for Standardization of Component BIST – Project, Phase 3**

   Commit a Verification Study to verify facts known about current existing IEEE Chip and Core test and debug standards. This includes understanding the access mechanism and architecture, hardware description, and vector file content that may be passed on to the board/system professional to facilitate IC BIST for Board/System Test, Characterization, and/or Debug. The following are likely recommendations that will be made following Phase 2 completion:

   a. Use IEEE-based standards to access chip-level features and functions to enable a board-test professional to meet board-test needs.
   b. The IEEE-based standards (1500, 1687) should be accessed through the 1149.1 TAP and TAP Controller.
   c. The on-chip embedded logic/functions/instruments should be “wrapped” and “accessed” by the appropriate standard:
      1) A complex IP such as a processor core should be wrapped with 1500 that has Local Instructions
      2) A less complex IP such as an instrument for temperature-voltage monitoring should be wrapped with 1687
3) When a few embedded IP are supported or very simple IP is supported, then 1149.1 TDRs may be sufficient
d. These three standards (1149.1/6, 1500, 1687) should cover all items of board test we are interested in
e. The impact to the iNEMI effort of this recommendation — is to limit the standard access of a chip on a board to the 1149.1 TAP

The key item for iNEMI on this recommendation is that board test professionals will expect to receive description files and vectors for board test that include the description files for all of these standards (and may be in one chip) -- 1149.1 BSDL and SVF, 1500 CTL and STIL, and 1687 ICL and PDL. Therefore, the team may

1) Make a recommendation on the chip-level access port (TAP)
2) Need to understand the languages that will need to be processed by the board-test user
3) Have to make a recommendation on how the languages associated with the IEEE standards should be used
4) Describe how a new board-level language should tie the existing standard languages together

**Program Background and Definitions**

**BIST Background**

BIST, as applied to Semiconductor IC’s, 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:

1. HSIO-BIST for characterizing the at-speed signal-integrity of the traces on the board.
2. 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.
3. External Memory BIST that tests an external memory that interfaces to another chip, as well as the interface itself.
4. The LBIST for determining if a failing board test is due to the board or is the fault of a specific chip.
   a. 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).
   b. 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.
   c. 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.
   d. 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-Add

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).

* 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), whereas 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.
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**

1. **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.

2. **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.

   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 re-converted 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 B
- Chip: (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:
   a. 1149.1 BSDL (pins, instructions, registers) and SVF (ScanDRs through selected JTAG Registers)
   b. 1500 CTL (cores, instructions, registers) and STIL (Vector Data associated with Cores)
   c. 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:

1. Chip A=Object-A; Chip B=Object-B
2. Object-A Transmits to Object-B Data-A that verifies the signal integrity of the Board Traces between them
3. Object-B Transmits to Object-A the acknowledgement that it received Data-A and that Data-A is correct
4. Data-A is defined to be 488-bit long packets in the ABC-Protocol
5. Then there is under this, a description of the Object-A and Object-B resources and requirements needed for the interaction of:
   a. Object-A uses P1687 Instrument-D that is described in ICL File-E and PDL File-F
   b. Object-B uses P1687 Instrument-G that is described in ICL File-H and PDL File-J
   c. Object-A PDL routine is named Start-Trace-Verification
   d. Object-B PDL routine is named Start-Receive-Data
   e. PDL Start-Trace-Verification and Start-Receive-Data must be asserted simultaneously

iNEMI BIST Use Case Investigation Project, Phase 1

Basic Project Information

Purpose of Project

Investigate and identify a use case to use as discussion, modeling, and development material including users of BIST (NPI, BT, BD&Y, RMA) and to identify the value-adds (what does BIST provide for the users specified).

Dependency

This work is predicated on the assumption that there is sufficient data available in the industry to make the decisions required within the scope of this project. If it is found during the course of this project that additional data is required, this SOW would be modified to include a test phase. The new SOW would be submitted to the iNEMI Technical Committee for approval prior to committing the project team to the tasks identified in the new SOW.

Business Impact

Currently IC BIST is generally only used at the IC level however; its application extends itself to board and system test and even in the field for remote diagnostics. The Phase 1 survey results show the industry is ready for this extended BIST usage.

BIST adds increased coverage and diagnostic capability. In addition, reduces test time and cost at board and system testing. It provides added diagnostic capability to isolate and troubleshoot field issues as well.

The project team will promote BIST usage throughout the component and product life cycle to the industry.
**Scope of Work**

1. Identify a use case to use as discussion, modeling, development material
2. Identify the Users of BIST (NPI, BT, BD&Y, RMA)
   - New Product Introduction
   - Board Test environments
   - BD&Y
   - RMA
3. Identify value-adds (what does BIST provide for the users specified)
   - List technical value add over traditional test methodologies.
   - List areas of cost savings
   - List test time savings
   - List manpower savings
   - Create paradigm shift of benefit of BIST

**Prospective Participants**

Agilent Technologies  
Alcatel-Lucent  
ASSET InterTech  
Cisco  
Corelis  
Dell, Inc.  
Delphi Electronics & Safety  
Flextronics International  
Hewlett-Packard Company  
Intel Corporation  
Lenovo  
Quanta Computer  
Teradyne, Inc.

<table>
<thead>
<tr>
<th>Task</th>
<th>iNEMI BIST Use Case Investigation Project</th>
<th>Quarters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Use Case 1 (chip to chip BIST)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Develop a use case to use as discussion, modeling, development material</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Identify the users of BIST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Identify value-adds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 White paper development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 End of project webinar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 BIST Use Case Function Classification Project, Phase 2 SOW Development</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Methodology and Resources

Phase 1 – Detailed Information

Resources: Team members from participating companies

1. Develop a use case to use as discussion, modeling and development material
   a) Identify the users of BIST
   b) Identify value-adds
2. Develop project white paper
3. Hold end-of-project webinar
4. Update BIST Use Case Function Classification Project, Phase 2 SOW – based on project outputs

Project Monitoring Plans

This project falls under the general category denoted as a Standards Development, i.e., given a set of materials and/or processes, these projects define a usable range for each set. The projects would also identify an appropriate standards body to which a proposal could be submitted to make the sets part of the published standards. The purpose of specification projects may also be to prepare “white papers” for industry distribution with the ultimate goal of making new specifications into de facto standards.

Project monitoring plans are as follows:

- Ensure open lines of communication among participants
  o Weekly conference calls
  o Meeting minutes provided through e-mail
  o Follow-up with individuals on an as-needed basis
  o Workshops and face-to-face meetings as appropriate
- Mid-project technical review and progress reports at regularly scheduled iNEMI meetings
- Track and document approximate Man-Months per quarter per team member (this will require the active members of the team to provide estimates).
- Track and document approximate number of people on the project per quarter (this can be tracked through iNEMI's WebEx account.)
- Project results, including conference presentations, technical papers, end-of-project webinar, etc., will be published on the iNEMI website.

Outcome of the Project

- Facilitate the development of new industry tools, processes and specifications
- Technical paper/whitepaper
- BIST Use Case Function Classification Project, Phase 2 SOW
General and Administrative Guidelines

General and Administrative Guidelines for this project and all other iNEMI Projects are documented at http://thor.inemi.org/webdownload/join/gen_guidelines.pdf.