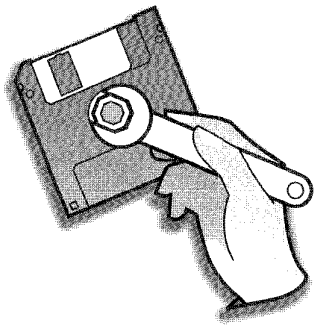

Built-In Self-Test: Assuring System Integrity

Bernd Könemann
Ben Bennetts
Najmi Jarwala
Benoit Nadeau-Dostie
LogicVision

Today's complex electronic products are harder to test using traditional external methods. BIST can frequently be used without significantly increasing a product's size, cost, and production time.

Microelectronics products and their components are tested many times during their life cycles. Integrated circuits begin their life on a wafer and are tested even before the wafer is diced into chips. The chips subsequently are packaged and used in higher level electronics. Each manufacturing step can introduce new defects that require new testing.

Each stage of the assembly process poses a different challenge for accessing, testing, and verifying chips and other product elements.

Accessing deeply embedded complex functions in higher level products to perform system-level tests can be very difficult. System-level integration turns each chip into a hierarchical assembly containing an ever-growing number of embedded core functions, such as memories, processors, and multimedia.

Determining the effects of the manufacturing process and environmental conditions on a product's ac and dc characteristics, debugging, and verification present additional problems. After all, it is not easy to probe submicron signal lines on buried wiring layers in the middle of a chip with a logic analyzer or an oscilloscope.

Also, limitations on bandwidth and on the performance of both the test equipment and the interface between the equipment and the product being tested make access difficult. Low-cost interfaces, like the ones used for system-level test access, tend to be serial and thus too slow for large amounts of test data. On the other hand, it is very difficult for automatic test equipment to access chips on a wafer through complicated high-performance interfaces.

Moreover, testing for calibration, stability, and noise is becoming more difficult to tackle as chip technologies begin to operate in the gigahertz operating frequency range. It is difficult to maintain clean waveforms across the interface between the tester and chips operating at such high frequencies.

Finally, regenerating and managing huge amounts of chip test data for each new assembly level, hardware revision, and system configuration quickly become a major burden on engineering resources and product development time.

In many cases, built-in self-test (BIST) alleviates these problems. And the addition of BIST features to electronics hardware frequently does not significantly increase a product's size, cost, and production time, as was the case in the past. This makes BIST practical in many cases.

ADVANTAGES OF BIST

In the 1980s, a number of system houses developed BIST solutions for distributing and embedding critical test functions into product components, in part because of data volume and access issues. The system houses did this because BIST avoids the complexities of external testing and access mechanisms by moving critical test and measurement functions inside products.

Embedded test and measurement utilities match the chip's technology capabilities and open up new possibilities for high-speed access to internal functions. In addition, BIST reduces the need for complex and expen-

0018-9162/96/$5.00 © 1996 IEEE

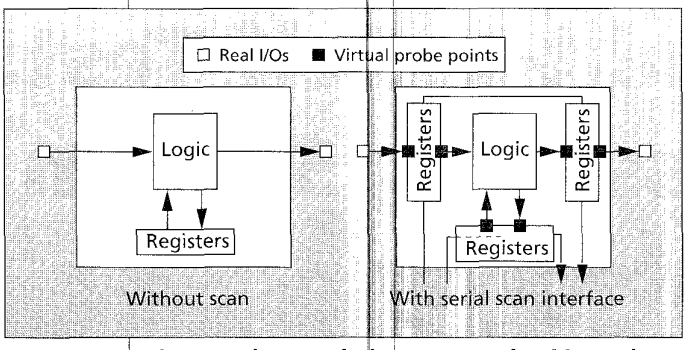

Figure 1. The scan design concept. The chip on the left is being tested without scanning. The product on the right, which is being scanned, has an embedded internal scan register on the bottom and two boundary scan registers on the sides, which are directly associated with the I/Os. By using the scan interface on the bottom of the product on the right, testers can communicate with the logic structure and the embedded registers without having to contact all of the chip's I/Os. This makes testing much more economical.

sive external test equipment. Unlike external equipment, BIST stays with the chip throughout its life cycle. The external BIST interface can be kept so simple that the test functions can be accessed, controlled, and reused economically at all assembly levels.

Meanwhile, BIST's cost is decreasing. Prior to the current generation of semiconductor technologies, adding comprehensive BIST features to a chip was expensive because it increased the size of the chip and it required a considerable design effort. Now, semiconductor technology lets you implement comprehensive chip-level BIST features for very little additional circuitry-related cost, while hardware synthesis technology for BIST integration has caused design-related costs to drop.

CORE BIST TECHNOLOGIES

Economy dictates that the BIST-related stimulus-generation and response-processing functions that we encapsulate into product components be very compact. In addition, adding BIST to a design must be easy and efficient.

Scan

Scanning is an established method for gaining access to electronic products' internal registers. Scanning involves the serial loading of arbitrary data into embedded registers from an external interface and the serial unloading of embedded registers' contents to an external interface. Users value scan design not only because it simplifies test generation, test application, and diagnostics, but also because it is an invaluable resource during hardware debugging and verification[1,2] and in many other system control and support functions.

The most popular scan style connects all or some of a product's registers into serial shift registers, also called *scan chains*. This lets you observe and/or control the register contents by serially loading and unloading scan chains. You exercise the logic between the scan chains by loading stimulus data into the scan chain cells and letting the values propagate through the logic.

You capture the responses into the receiving register cells, from which you unload them for comparison with the expected response values. Because scanning lets you observe and control register values, as if you attached physical probes to each register, the scan chain cells act as virtual probe points.

Scanning permits the observation and control of many register bits from one input pin, at one end of the scan chain, and one output pin, at the other end. Each scan interface thus requires only two data pins, rather than one pin for each of the chip's pins. Scanning, therefore, is economical for low-cost, low-bandwidth systemwide data communication between hardware components and a test processor, such as a system service processor.[2] However, scanning is also slow because data must be taken from a scan chain one bit at a time.

Sometimes you add special scan cells to control and observe nets that are not functional registers. Boundary scan, for example, adds scan cells to the product's signal I/Os. The virtual probe points that boundary scan adds to the I/O structures make it possible to exercise the internal logic without involving the I/Os (internal test), as shown in Figure 1, and the I/Os without involving the internal logic (external test). This makes the process less expensive and more reliable.

Test access port

The IEEE 1149.1 standard for test access port and boundary-scan architecture[3] defined an industrywide interoperability framework for boundary scan, originally for the purpose of testing the interconnection between chips on higher level assemblies, such as printed circuit boards.

Traditionally, you conduct interconnection testing by physically probing and verifying each interconnection net. With the IEEE 1149.1 boundary scan method, you can conduct the same type of test with much simpler fixtures and equipment by using a four-wire or five-wire serial interface called the test access port (TAP). The port, as shown in Figure 2, has a standardized, well-defined instruction architecture and communications protocol.

The IEEE 1149.1 standard permits user-defined instructions, as well as several mandatory and suggested instructions. This, together with the small footprint, turns the test access port into a flexible, powerful, and economical, albeit slow, industry-standard interface for communicating with a chip and its built-in hardware utility functions. And because the interface does not have to contact all I/Os, it can be even less expensive.

TAPs that conform to IEEE 1149.1 can already be found in many commercial chips, including many popular microprocessor families. They are also becoming increasingly popular in application-specific integrated circuit (ASIC) designs. A growing number of interface equipment types and software tools that address a wide range of applications support the standard. Some of the applications, such as the programming of field-programmable gate arrays and the emulation of processors, make the standard even more useful than originally intended.

BIST of logic structures

BIST of logic provides low-cost stimulus-generation and response-processing for testing complex logic structures.

STIMULUS GENERATION. Pseudorandom pattern generators (PRPGs) have emerged as a popular choice for stimulus generation.[4] PRPGs are finite-state machines that produce long data sequences from an initial seed value.

PRPG sequences are totally predictable and repeatable, which makes it possible to simulate test stimulus sequences and expected signatures. However, PRPG sequences appear random, so they provide a variety of stimuli that test all parts of a product equally. On the other hand, using a binary counter, for example, would stimulate the part of the product driven by low-order bits much more than those driven by high-order bits. Practical PRPGs tend to have 16 or more state bits to assure sufficiently long test stimulus sequences that do not repeat before the PRPG returns to the seed state.

RESPONSE PROCESSING. Signature analysis has become a key response-processing technology.[4] Tests of complex chips can produce many megabytes of response data. You need to compare this data with data you expect to receive from a defect-free chip. However, it is not cost-effective to store many megabytes of such comparative data on a chip.

A signature reduces the information that needs to be compared to just a few bytes (4 bytes if a 32-bit signature is used). A signature is a short code word that captures certain characteristics of the test-response data sequence. The code helps detect errors, as any errors in the test-response sequence will very likely change the derived signature. It is thus sufficient to compare the signature that represents the test-response sequence with the signature of the error-free response sequence of a defect-free product.

Signature-generation hardware that can process several response bits in parallel is called *multiple-input signature register* hardware. MISR hardware typically uses a structure consisting of a register bit (such as a flip-flop) and a number of exclusive-OR and other gates for each code bit.

Occasionally, error information gets canceled during code generation, in which case the final signature looks like an error-free signature, even though there were response errors. This is called *aliasing*. A large body of research and practical experience[4,5] suggests that the probability of aliasing in an n-bit signature is one in 2^n. Typical signatures used in BIST have at least 24 bits. The probability of aliasing in a 24-bit signature is one in 16,777,216.

SCAN-BASED BIST OF LOGIC. As shown in Figure 3, BIST of logic structures can be combined with scanning methods by connecting a PRPG to the inputs of the scan chains and an MISR to the outputs. The interface between the PRPG/MISR and the scan chains is contained on a chip and thus can have many short scan chains in parallel, thereby achieving high internal data bandwidths.

An on-chip controller for BIST of logic minimizes external data requirements. This controller generates all the control and clock waveforms necessary to repeat the scan's loading and unloading operations and capture cycles until

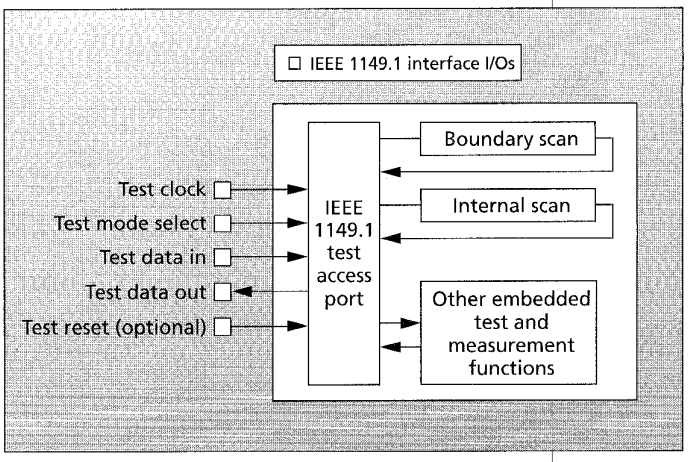

Figure 2. IEEE 1149.1 Test access port (TAP). Using the IEEE 1149.1 boundary-scan method, you can conduct interconnection testing more easily than the traditional method, which required physical probing. To download instructions to embedded chip functions, or to retrieve status information from them, you use the TAP, a powerful, although slow, serial interface. The interface does not have to contact all of the chip's I/Os, so it can be inexpensive. A typical TAP's four input pins and one output pin are shown on the left.

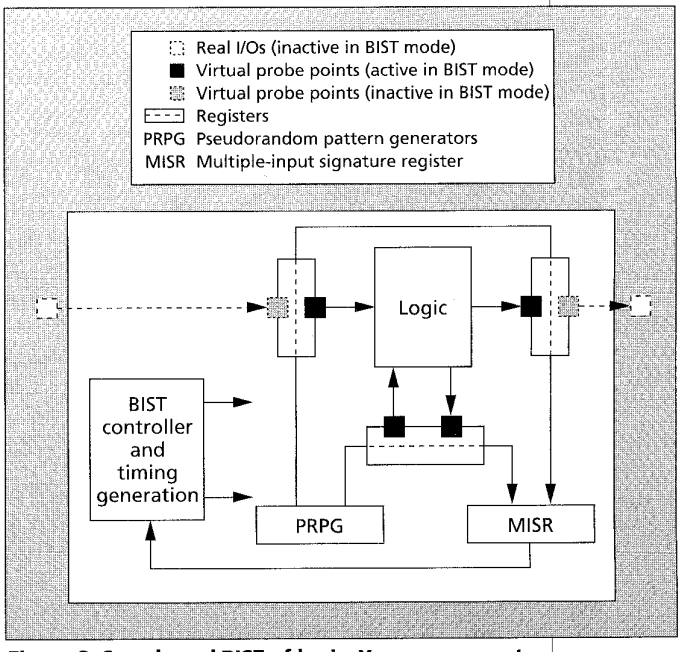

Figure 3. Scan-based BIST of logic. You can connect a PRPG to the inputs of the scan chains and an MISR to the outputs. After the test stimulus has propagated through the logic, responses are captured into the registers and sent to the MISR. The MISR uses such data to make a signature, which contains the test results that need to be compared to the results you expect from a defect-free product. The BIST controller makes sure these steps are repeated until the test is finished.

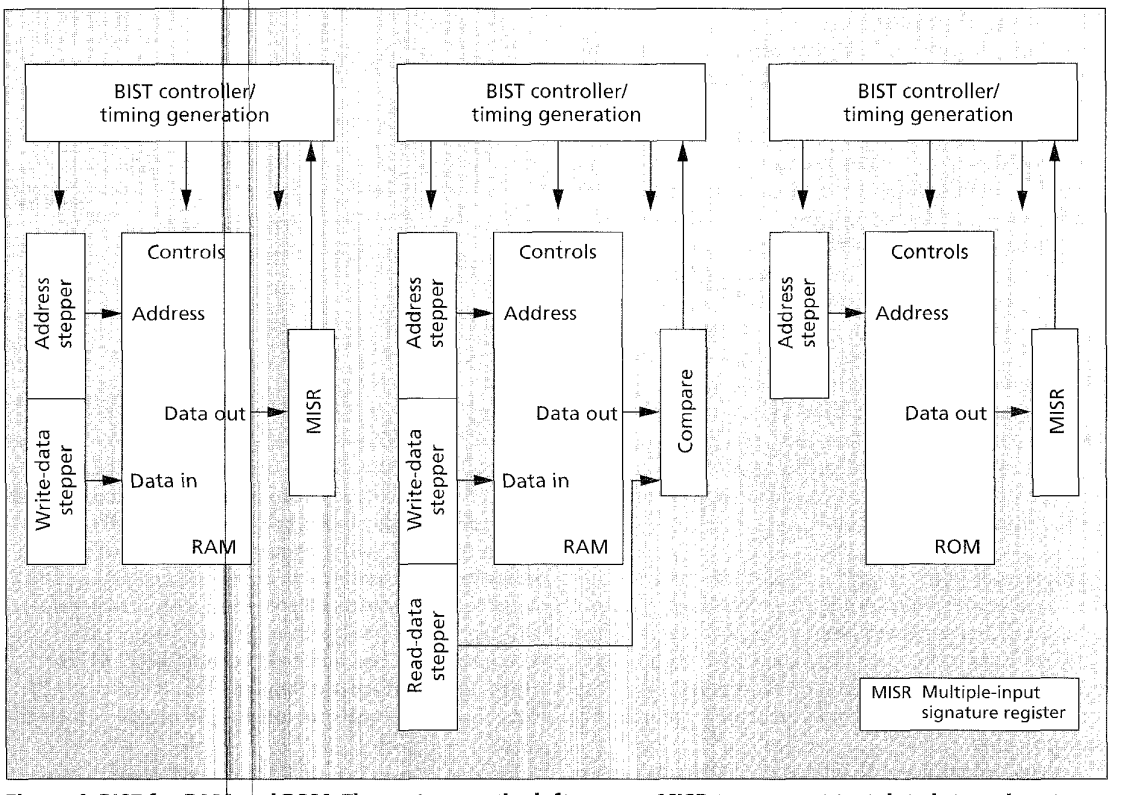

Figure 4. BIST for RAM and ROM. The system on the left uses an MISR to compact test data into a signature. Testing is much easier when you can use a small amount of data to compare test results with those expected from a defect-free product. The system in the middle, on the other hand, demands on-the-fly generation of data expected from a defect-free product. This makes the data available for comparison every time a read operation is performed, which lets you test a memory while it is operating at its maximum specified speed. The system on the right is similar to the system on the left, but because a ROM does not have write-data inputs, it does not need a write-data stepper.

the test is completed and the final signature is ready for comparison. The controller may have its own command and status registers to initiate various test modes and/or enable diagnostic operations.

With virtually all critical BIST signals and data generated and distributed on the chip, BIST will become an increasingly attractive platform for testing at normal clocking rates as chips get faster and external interfacing becomes extremely difficult.

Since the 1980s, scan-based BIST of logic has been used extensively for some very complex chips and higher level assemblies, such as MultiChip Modules (MCMs) and printed circuit boards.

Many complex logic structures detect at least 90 percent of faults with pseudorandom patterns. Detection is limited by faults that are resistant to random-pattern testing. For a higher detection rate, insert a few test points (additional logic connections that improve testability by making nets easier to control and/or observe) into the logic to reduce or eliminate the random-pattern resistance.[5,6]

One interesting side-effect of random-pattern testing is the detection of nontarget defects. Unlike external testing, LogicBIST can economically apply many more tests than are required for detecting targeted faults. This makes it more likely you will detect unanticipated defects.[5]

The controller for BIST of logic can be connected to an on-chip TAP to take advantage of the IEEE 1149.1 interface and communications protocol. The slow, low-cost TAP interface is suitable for communicating with the controller for BIST of logic because only minimal external data traffic is involved.

BIST of memory

Practically all complex chip designs contain embedded RAM or ROM. Modern telecommunications chips can have dozens of embedded memories. In addition, multiport memory macros with hundreds of I/Os are becoming more popular.

Memories are known to fail in ways that require special, memory-oriented tests. Applying these tests externally to the embedded memories is difficult for the same reasons external testing of other core functions is difficult. BIST of memory, shown in Figure 4, eliminates the need for external access paths and enables test reuse by providing built-in stimulus-generation and response-processing utilities for embedded-memory testing.

STIMULUS GENERATION. Memory-test stimulus sequences tend to be highly regular, structured, and algorithmic. March algorithms[7] are exceptionally simple and

are very good at detecting defects. They make several forward and backward passes through the memory address space to "march" a new data pattern over the pattern that was left in the memory from the previous pass.

RESPONSE PROCESSING. There are two response-processing options for RAMs. You can use an MISR to compact the read-data sequence into a signature, or you can compare each read-data value with the value expected for a defect-free memory. The March algorithms write very simple and regular data patterns into a RAM, which makes it possible to efficiently generate the data expected for a defect-free memory on the fly in a read-data stepper. A *read-data stepper* is a small finite-state machine that generates the expected RAM output data for read operations. A *write-data stepper* generates the expected RAM input data to be used in write operations.

The data patterns stored in ROMs, on the other hand, cannot be predicted algorithmically. Hence, it is impossible to develop a simple finite-state machine that could generate the data expected from a defect-free memory. Most BIST implementations for ROMs thus handle data by using an MISR to compact test responses into a signature.

As with BIST of logic, it is preferable to drive BIST of memory with an on-chip controller that executes the test algorithm and generates all high-speed control and clock timings for testing at normal clocking rates. Controllers for BIST of memory can have their own command and status registers that permit different test modes and/or diagnostics. BIST of memory controllers, like BIST of logic controllers, require little external data communication and can be connected to an on-chip TAP.

BIST design

Because companies want to produce product designs as quickly as possible, the effort required to add BIST is a critical issue. Previous generations of BIST had to be designed primarily by hand, which is difficult, time-consuming, and tedious. Today, chip designers take advantage of advances in synthesis technology. This permits the rapid implementation of BIST hardware functions from high-level specifications at the same time as and at the same abstraction level as the functional design implementation. The workflow for designing hardware with BIST is shown in Figure 5.

Modern BIST design tools integrate BIST functions with the rest of the chip design at the register transfer level, as part of the front-end design process. This greatly reduces the burden on the designers and avoids costly back-end design iterations.

Implementation choices are driven by simple BIST configuration scripts that you can create without BIST expertise. The BIST tools automatically select, configure, instantiate, and connect appropriate elements from a growing library of register-transfer-level BIST design objects, such as controllers. The tools also generate simulation test benches for verifying the generated structures, as well as scripts for synthesizing gate-level implementations of the register-transfer-level BIST objects and the rest of the design.

Although the effort to bring all BIST design functions to the register transfer level is progressing, designers still must perform some back-end work at the gate level. For example, fault simulation relies on gate-level fault models.

TAKING BIST TO THE NEXT LEVEL

BIST's ultimate value will be determined by how well the technology can grow and by the benefits derived when it is used throughout the development process in products of increasing complexity.

Chip-level tests

BIST can be used to conduct chip-level manufacturing tests. Combining BIST and boundary scan, you can thoroughly test even the most complex high-pin-count, high-performance chips using a low-pin-count interface. This eliminates the need for expensive test equipment and probes. Meanwhile, researchers are working on ways to add built-in utilities for measuring continuous parameters, including such electrical specifications as dc parametrics (for example, leakage currents) and such timing specifications as ac parametrics (for example, chip I/O delays). Researchers are also working on built-in utilities that would permit the testing of mixed-signal (analog and digital) functions with testers currently used just for digital functions.[8]

Higher level tests

Complex electronic products must be reliable, available, and serviceable in the field. Consequently, many products contain extensive hardware test and diagnostic support functions that can be executed in the field.

The development of higher level hardware diagnostics is greatly simplified by encapsulating comprehensive tests into each chip.[9,10] A diagnostic system can call local BIST functions and receive each chip's results with little

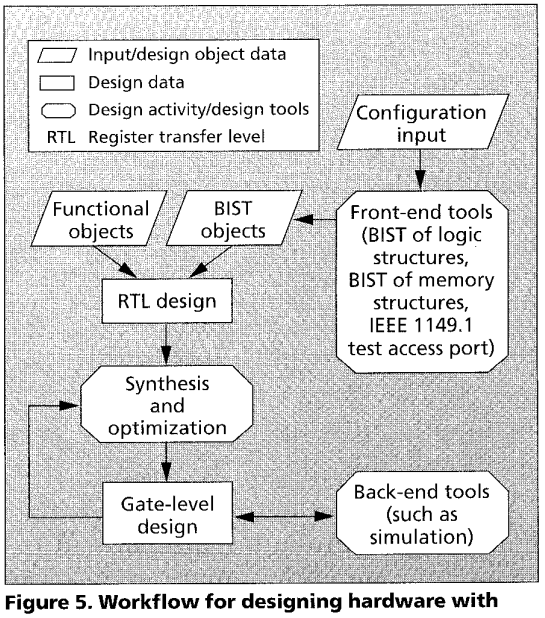

Figure 5. Workflow for designing hardware with BIST. BIST-related design objects are combined with other design objects early in the process, and they proceed together through the normal hardware design flow.

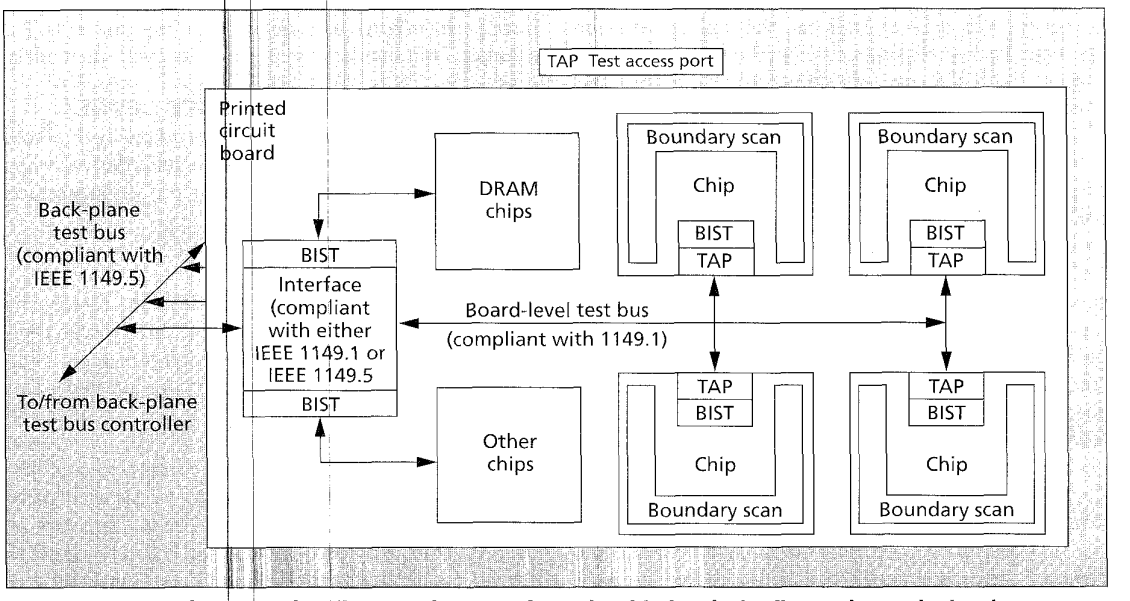

Figure 6. Concept for hierarchically extending BIST from the chip level. The figure shows the key bus connections and design elements that are needed to use and access embedded BIST functions across a typical system assembly. The interface/bus controller on each circuit board includes mechanisms for addressing the scan registers and BIST functions on each chip and for receiving and sending messages on the external bus. That means a service processor attached to the external bus can connect with each circuit board and then use the internal bus to access each chip and its embedded scan registers and BIST functions.

data-related and communication-related overhead. This will occur as long as diagnostic systems can access BIST functions via, for example, hierarchical BIST interfaces.

Access is thus an important issue for BIST technology. The strong acceptance of the IEEE 1149.1 standard for board-level and chip-level access has been an excellent starting point. Meanwhile, several proposed standards for extending the TAP interface to the module, backplane, and system levels, including IEEE 1149.5,[11] are under consideration.

Many products, such as electronic assemblies containing stand-alone memory chips without chip-level BIST, can benefit from higher level BIST functions. Stand-alone chips often have dynamic RAM. In addition, board-level and system-level memory frequently is available in several configurations and can be upgraded in the field. Thus, board-level and system-level BIST of memory cannot simply copy existing chip-level BIST of memory but instead must understand stand-alone memory chips' complex access modes and must be flexible enough to accommodate different memory configurations and upgrades.

Figure 6 illustrates the concept of hierarchically extending BIST from the chip level.

TRADITIONAL EXTERNAL TEST EQUIPMENT HAS a hard time accessing core hardware functions that are embedded more and more deeply and that are increasingly complex. Chip technology's steadily increasing operating frequencies also challenge test equipment and interfacing technologies. BIST, however, has met these challenges. And, in terms of increased chip area and design effort, the cost per logic gate of embedding comprehensive BIST functions into integrated circuit chips has become affordable to many.

We now must focus on applications that would encourage the electronics industry to use BIST extensively. This would better enable BIST to help assure the integrity of complex electronic products and to help keep innovation in electronics technology affordable. ■

References

1. H. Hao and R. Avra, "Structured Design for Debug—The SuperSPARC II Methodology and Implementation," *Proc. Int'l Test Conf.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1995, pp. 175-183.
2. H.W. Miller, " 'Design for Test' via Standardized Design and Display Techniques," *Electronics Test*, Oct. 1983, pp. 108-116.
3. K.P. Parker, *The Boundary Scan Handbook*, Kluwer Academic Publishers, Norwell, Mass., 1992.
4. P.H. Bardell, W. McAnney, and J. Savir, *Built-In Test for VLSI: Pseudo-Random Techniques*, John Wiley & Sons, Somerset, N.J., 1987.
5. E.B. Eichelberger et al., *Structured Logic Testing*, Prentice-Hall, Englewood Cliffs, N.J., 1991.
6. B.H. Seiss, P.M. Trouborst, and M.H. Schulz, "Test Point Insertion for Scan-Based BIST," *Proc. European Test Conf.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1991, pp. 253-262.
7. A.J. Van de Goor, *Testing Semiconductor Memories: Theory and Practice*, John Wiley & Sons, Somerset, N.J., 1991.
8. S. Sunter, "The P1149.4 Mixed Signal Test Bus: Costs and Benefits," *Proc. Int'l Test Conf.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1995, pp. 444-450.

9. H. Hulvershorn, P. Soong, and S. Adham, "Linking Diagnostic Software to Hardware Self-Test in Telecom Systems," *Proc. Int'l Test Conf.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1995, pp. 986-993.
10. E.C. Behnke, "3B21D BIST/Bscan System Diagnostic Test Story," *Proc. Int'l Test Conf.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1994, pp. 120-126.
11. R.E. Tulloss, "Leave the Wires to Last—Functional Evaluation of the IEEE Standard 1149.5 Module Test and Maintenance Bus," *Proc. Int'l Test Conf.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1995, pp. 797-806.

Bernd Könemann *is senior director at LogicVision in San Jose, California. He has worked in the areas of BIST, design for test, the development of test-generation and test-application methodology, and the architecture of test-support tools. He has written more than 20 technical publications. He received a doctorate in theoretical physics from the Technical University in Braunschweig, Germany.*

Ben Bennetts *is market development director for LogicVision in the UK. He has worked in various test-related areas, including design for test. He was a member of the Joint Test Action Group, which delivered what is now the IEEE 1149.1 standard for boundary scan. He has written three books and more than 75 articles on subjects relating to testing and design for test. He received a PhD from Southampton University, UK.*

Najmi Jarwala *is chief technologist at LogicVision in Princeton, New Jersey. He has worked on the development and deployment of test technology for boundary scan, and on projects relating to BIST and system testing. He has also served as chair for the IEEE 1149.1 Boundary Scan Standard Working Group. He has written more than 35 articles for IEEE journals and conferences. He received a PhD in electrical and computer engineering from the University of Massachusetts.*

Benoit Nadeau-Dostie *is chief scientist for LogicVision in Ottawa, Canada. He has worked in the areas of BIST and microelectronics design for test. He has written numerous articles for journals, magazines, and conference proceedings about BIST of memory, logic, and printed circuit boards. He is on the editorial board of* IEEE Design & Test of Computers, *and was elected to be a senior member of the IEEE. He received a PhD in electrical engineering from the Université de Sherbrooke in Quebec, Canada.*

Contact the authors at {berndk, benb, jarwala, benoit}@lvision.com.