BOUNDARY-SCAN TESTING

Boundary-Scan is a collection of design rules which, when implemented in digital integrated circuit (IC) chips, allow the chips to test faults occurring when these ICs are assembled on printed circuit boards (PCBs) and in systems. The Boundary-Scan rules are formally presented in a document, IEEE/ANSI Standard 1149.1a-1993 (1), “A Standard Test Access Port and Boundary-Scan Architecture.” A Boundary-Scan compliant IC is one that complies with the design rules in this document. Boundary-Scan is sometimes referred to as 1149.1 and also as JTAG. JTAG refers to the Joint Test Action Group formed in 1985 by major electronics firms around the world to address a growing problem in board testing. Between 1986 and 1988, the JTAG Technical Subcommittee developed many proposals, resulting in approval of the IEEE 1149.1 Standard in 1990. This standard was subsequently modified in 1993 to address implementation concerns.

It is important to realize that the 1149.1 Standard is only a document which describes the basic rules to which a Boundary-Scan architecture must comply. While it includes possible embodiments for various Boundary-Scan components, it does not define preferred implementations. Specific architecture-related details are user-defined (2). It is also important to realize that while the 1149.1 Standard is generally identified with Boundary-Scan, other approaches are possible. This article begins by indicating the problem which Boundary-Scan was designed to address, followed by a description of the basic Boundary-Scan structure, issues in the application of Boundary-Scan, and emerging extensions of the Boundary-Scan concept.

PROBLEMS IN BOARD TEST

Boundary-Scan was developed to solve problems in testing digital IC-populated printed circuit boards. Many defects, such as open circuits at pins or shorted traces, can occur during the manufacturing process. Testing to discover these defects is greatly simplified if one has direct access to internal test points such as IC pins. If test points are only available at the PCB’s edge connector, it can be quite difficult to activate a fault from primary inputs (the controllability problem) and propagate the effect of the fault to the board's primary outputs (the observability problem). Besides detecting a fault, it is often very difficult to provide an accurate fault diagnosis.

Conventional, one-sided PCBs were populated with integrated circuits in dual inline packages (DIPs) having 0.1 inch = 100 mil ≈ 2.5 mm pin spacings. Several layers of conducting traces on the PCB were used to interconnect the pins of ICs and other components. These pins extended through the board (through-holes in the PCB) and were soldered on the other side of the board. This allowed access to the IC pins by a bed-of-nails fixture contacting the back of the PCB in an in-circuit tester or by a hand-held probe for fault location. This is not practical in modern electronic manufacturing. Surface mounted IC packages do not provide access to personal computer (PC) backs (which may also be populated with components). Moreover, the distance between pins is shrinking to 25 mils for surface mount devices (SDMs) and 12 mils for tape automated bonding (TAB), making contacts increasingly difficult. The result is a radical decrease in test point accessibility. This problem is amplified for more recent package technologies such as Ball Grid Arrays and Chip-Scale Packaging.

Boundary-Scan is a design-for-testability (DFT) philosophy. It aims to simplify testing by modifying ICs to improve the testability of IC interconnections on boards and to provide access to internal signals. Although the 1149.1 Standard provides support for internal DFT and Built-In Self-Test (BIST), internal DFT and BIST are not required for compliance with the standard. In contract to DFT, BIST allows in-circuit, at-speed testing while minimizing reliance on external testers by adding logic structures such as Linear Feedback Shift Registers (LFSR), signature analyzers, or Built-In Logic Block Observers (BILBO) (3).

Scan is a well-established technique for providing access to internal test points in large circuits. In scan-based testing, all the flip-flops and registers in a circuit are chained together to form a path between a primary input and primary output. For combinational logic between registers, scan allows test vectors to be scanned into the input register and test results to be scanned out of the output register. Boundary-Scan uses this technique to test the integrity of board interconnections between IC pins by chaining together scan cells associated with every I/O pin of the IC.

BASIC PRINCIPLES OF BOUNDARY-SCAN

"Boundary-Scan test is a structured design-for-testability method applicable to digital devices" (4). The essential idea behind Boundary-Scan is described by the term itself. The first term, *boundary,* indicates that the testability circuitry is placed at the I/O pins of a component, around the system logic. The second term, *scan,* suggests that the additional test circuitry within the component is transformed into a serial shift register during testing. The serial shift register is often referred to as the "scan path" and forms a controllable border around the design. Access to the test logic is provided via four or five (one optional) dedicated test pins. At the PCB level, the registers for the individual components can be connected in series to form a single path through the entire design. Or a board design can contain several independent paths.

Figure 1 shows the scan cells of several ICs chained together to form a scan path from board scan-in to board scan-out. A scan cell is associated with each input or output pin, and each can store a bit. That bit can enter the cell by parallel or serial (scan) paths and exit the cell by parallel or serial paths, depending on how internal switches are set within the cell. Typically, data passes in parallel through the cells in their normal operating mode and passes serially in their test (scan) mode.

Figure 2 shows the general Boundry-Scan architecture for an IC as prescribed in the IEEE 1149.1 Standard (1). We distinguish the following basic elements:

1. Test Access Port (TAP)
2. TAP Controller
3. Instruction Register
4. Test Data Registers [Boundary Register, Bypass Register, Device Identification Register (optional), Design-Specific Registers (optional)]

The TAP provides access to the test support logic built into the IC. It consists of three required input connections and one output connection. The optional fourth input connection is used to reset the Boundary-Scan test logic. The IEEE 1149.1 Standard requires that the TAP connections are not used for any other purpose than testing. The mandatory TAP connections are: (1) Test Data In (TDI), (2) Test Clock (TCK), (3) Test Mode Select (TMS), (4) Test Data Out (TDO).

The scan paths of several ICs are connected by wiring TDO from one chip to TDI of another. The other two reserved pins, TCK and TMS, are generally wired in common for all the ICs on a board so that the states of all Boundary-Scan components are changed similarly and simultaneously. The *optional* Test Reset (TRST) pin, if implemented, allows one to reset test logic on the component independently of the internal system logic. There is also a synchronizing sequence utilizing TCK/TMS that can be used to reset the logic. Since TRST is optional and fairly uncommon, this sequence will be used most often for rest. The set of scan cells associated with I/O pins can be connected to form a boundary (shift) register.

The TAP controller is a synchronous state machine that allows the test circuitry to load and execute instructions controlling scan cell operation. The controller responds to changes in the TMS and TCK signals and generates the clock and control signals required to shift, capture, or update data through either the instruction or test data registers (boundary register and bypass register). The Bypass Register is a single state shift register which allows data to bypass the scan path on an IC by directly connecting the TDI pin to the TDO pin. The instruction register is placed between TDI and TDO, just like the boundary register, so that test instructions can be shifted in. The test instruction defines the test to be performed and the test data register to be addressed. Each Boundary-Scan architecture has to contain the boundary and bypass test data register. The device identification and design-specific register are optional. Each test data register can be selected, in other words placed between TDI and TDO, by scanning the corresponding instruction code into the instruction register.

The state diagram for the TAP controller has two main loop sequences, one for handling test data and one for han-

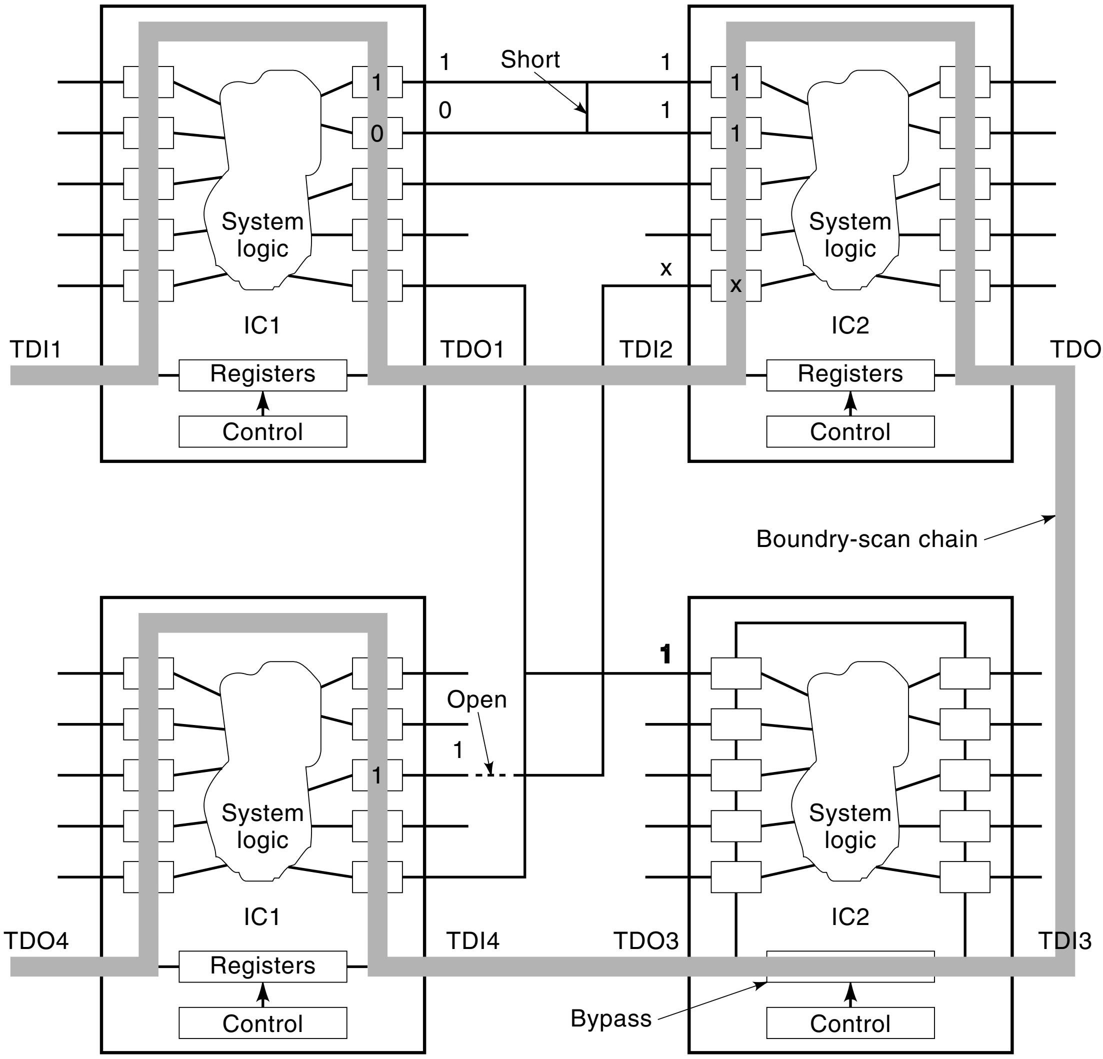

Figure 1. Boundary-Scan ICs on a PC board linked together in a Boundary-Scan chain for the testing of interconnecting wires between the ICs.

dling instructions. Figure 3 shows the loop for handling test data as defined by the IEEE Standard. Both loops follow an identical sequence, the difference being whether the TAP controller is interacting with a test data register or the instruction register.

In order to understand how Boundary-Scan operates as a test technique, the key instructions need to be discussed. All instructions are serially loaded into the instruction register and are decoded to achieve two basic functions: (1) select the test data register that can operate while the instruction is active, (2) control non-selected test data registers so that they do not interefere with the normal on-board operation of the particular IC. Several instructions are mandated for Boundary-Scan, and more are optional. Mandated instructions are External Test (EXTEST), SAMPLE/PRELOAD, and BYPASS. In addition, both optional instructions, such as Internal Test (INTEST), Run Built-In Self-Test (RUNBIST), HIGHZ and CLAMP, as well as user-defined instructions, such as IDCODE and USERCODE, are allowed.

EXTEST allows for the observation of the input cells and the control of the output cells of the Boundary-Scan components. It is the key Boundary-Scan instruction. It provides easy board-level interconnect testing of opens, stuck-at, or bridging faults, etc. This instruction also facilitates testing of components that do not support the Boundary-Scan test themselves but are surrounded by Boundary-Scan components. Test values are applied externally to the chip via the boundary register which at the same time isolates the on-chip system logic from those test signals used to test the integrity of external connections to the IC.

The SAMPLE/PRELOAD instruction effectively provides two functions. It allows a SAMPLE of the normal operation of a component to be taken for examination. Prior to the selection of another test operation, PRELOAD loads values into the boundary register. The BYPASS instruction sets up a single-bit route between the TDI and TDO pin providing a shortcut route between a component's TDI and TDO pin for board-level testing.

INTEST allows static, slow-speed testing of on-chip logic without affecting the operation of surrounding components on a PCB. As test data has to be applied serially to TDI, the apparent testing rate is greatly reduced. The reduction is directly proportional to the length of the boundary register. Test data is shifted into the boundary register and consequently applied to the on-system logic. Test results are then captured and shifted out for evaluation. Note that INTEST is an *optional* instruction. Internal testing of ICs can be facilitated with Boundary-Scan but is not guaranteed in a Boundary-Scan compliant IC. RUNBIST offers a supplementary method of assisting on-chip testing if the IC has BIST capability. It causes the execution of a self-contained self-test, without the need to load complex data patterns or a single step operation as for INTEST. The optional HIGHZ instruction targets the Bypass Register and allows one to place all outputs of the Boundary-Scan IC into a high impedance (high Z) state. It delivers in-circuit isolation by shutting off the inputs to other ICs on the PCB. Finally, the optional CLAMP instruction also targets the Bypass Register and forces fixed values on an IC's output pins without incurring the overhead of its entire Boundary Register. If the optional device identification register is included in the design, an identification code instruction (IDCODE) has to be included as well. It will provide additional information about the component. If the component is user-programmable, the user defined code (USERCODE) instruction has to be incorporated. It contains information about the programming of the component.

Figure 4 shows a possible implementation of a Boundary-Scan cell for unidirectional pins (2). Most Boundary-Scan cells contain a shift stage flip-flop and a parallel hold or update stage flip-flop. The shift stage flip-flop holds the test data moving through the boundary register. The update stage is updated from the shift stage. For extremely performance-sensitive component inputs, the Standard allows a monitor-only Boundary-Scan cell which omits the update stage. In this case, the cell design cannot support INTEST or RUNBIST, because the system logic cannot be isolated from the effects

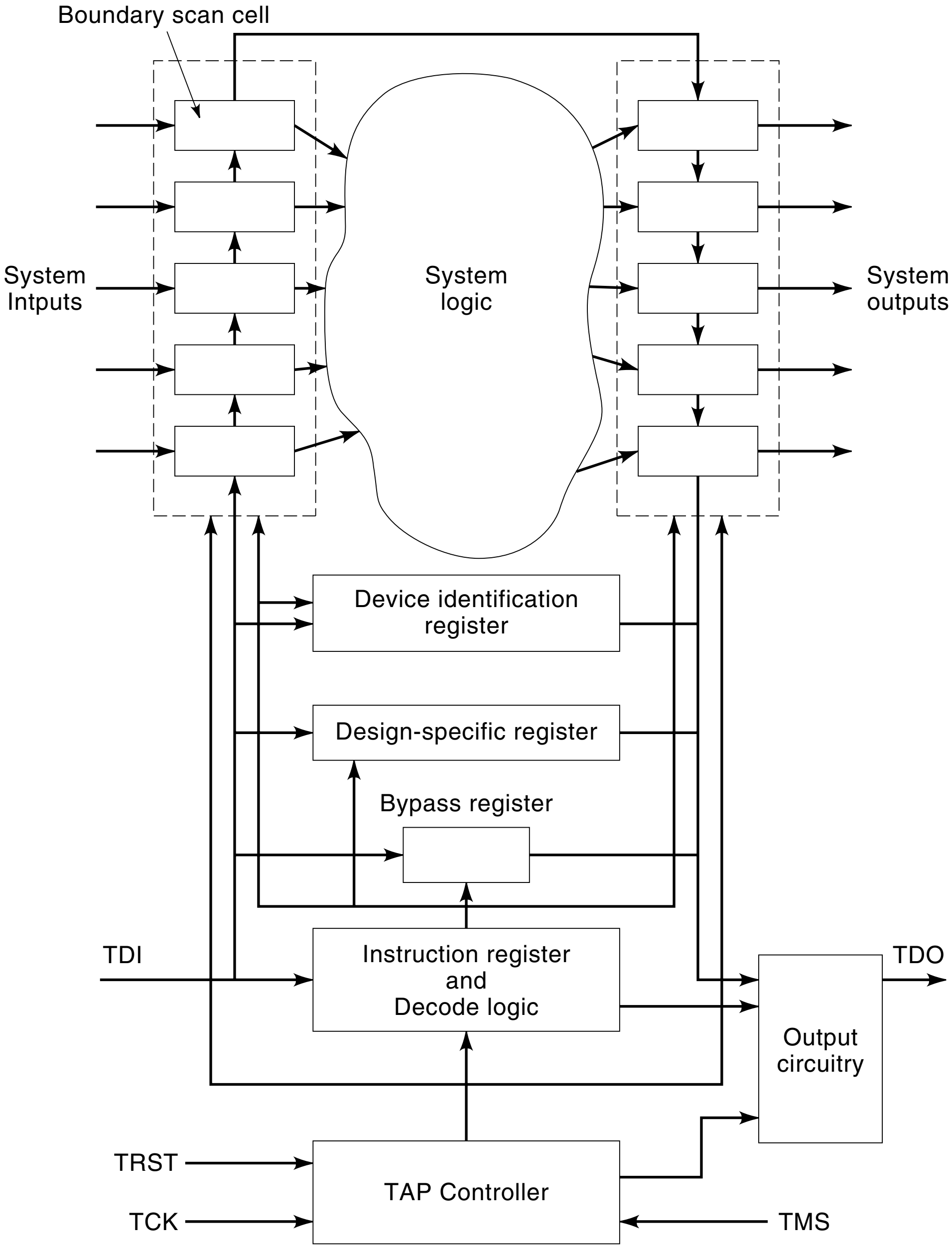

Figure 2. IEEE Standard 1149.1 Boundary-Scan architecture. The architecture basically consists of a Test Access Port (TAP), a TAP controller, a Boundary-Scan register, a Bypass register, an Instruction register, and possibly some user-defined test data registers.

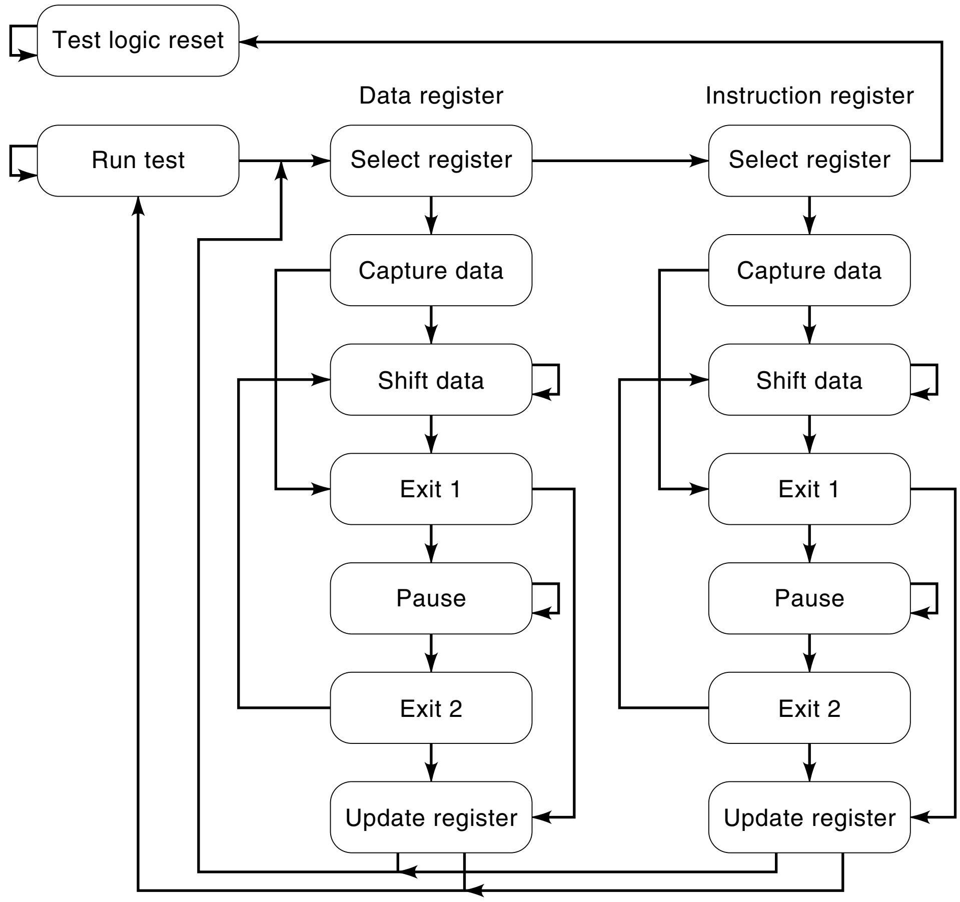

Figure 3. TAP controller state diagram. The controller is a finite state machine with 16 states. The two similar vertical columns of seven states reference data registers and instruction registers, respectively. They behave in an identical fashion.

of external signals. For Boundary Register support of bidirectional pins, two approaches are available. The first one uses two data register cells: one as an input and one as an output. The second one implements a single, reversible cell to perform both the input and output function. In both cases, a control cell is added that gives the Boundary Register control over the output enables of the drivers.

APPLYING BOUNDARY-SCAN TO BOARD TEST

Before using Boundary-Scan to check for interconnection faults, one needs to verify that the Boundary-Scan circuitry is present and working. First, one checks that the TDI, TMS, TCK, and TDO pins can be driven both high and low. Then, one checks that the boundary register, instruction register, and bypass registers of any Boundary-Scan IC can be loaded with combinations of 1s and 0s.

Once it is established that the Boundary-Scan circuitry is functional, EXTEST can be used to check for interconnect opens and shorts. To test the behavior of interconnects connecting the outputs of IC1 to the inputs of IC2 and IC4 (see Fig. 1), EXTEST is used to control the IC1 outputs (test vector) and observe IC2 and IC4 inputs (response vector). The interconnects to IC3 are not tested, and therefore, the Bypass register of IC3 is inserted in the chain. When there are no faults, the node patterns on the IC1 outputs will match those on the IC2 and IC4 inputs. Identical node values on IC2 or IC4 inputs suggest shorts (bridging faults) between the corresponding interconnects. Opens will cause IC2 or IC4 inputs

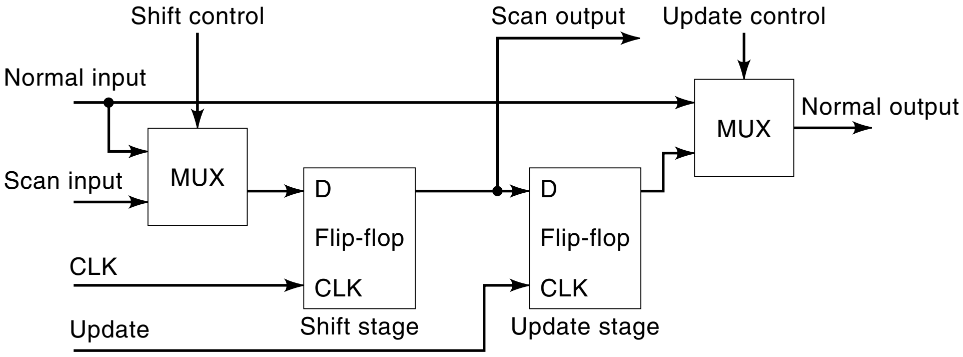

Figure 4. Example of a unidirectional Boundary-Scan cell. The pins labeled “Scan Input” and “Scan Output” are the serial inputs and outputs of the Boundary-Scan register. The regular input/output pins of the cell are labeled “Normal Input” and “Normal Output.” All other pins route control signals from the TAP controller to the cell.

to float high. Not all interconnect tests are as straightforward as those outlined above. ICs are usually connected in a more complicated fashion than described here. Faulty node values may match expected, correct node values unless both high and low values are checked.

In practice, detecting and diagnosing faults on board interconnects is usually very complex. Jarwala and Yau (5) have summarized the types of faults which can occur on boards and the effectiveness of various algorithms in detection and diagnosis. They consider both multi-net and single-net faults. Multi-net faults correspond to bridging faults which create a short between two or more nets. A net on a board corresponds to an equipotential surface which may connect multiple input drivers to multiple output buffers. Consequently, multi-net faults include OR-type (driver "1" dominates), AND-type (drive "0" dominates), and strong-driver shorts (one driver dominates). Single-net faults are stuck-at-1, stuck-at-0, and open faults on single nets. To clarify the issues in fault diagnosis, Jarwala and Yau (5) introduce a set of parallel test vectors applied to all nets in parallel, and the corresponding set of sequential test vectors, the vectors applied to particular nets over time by a number of parallel test vectors. The sequential test vectors for a faulty net provide a response which can be used for diagnosis. Effective fault diagnosis requires identification of which nets are shorted, open, stuck-at-0, or stuck-at-1. Fault diagnosis is complicated when the faulty response of one net is the same as the fault-free response of another net. In any case, interconnect fault detection and diagnosis require a careful choice of the test patterns or vectors to be scanned into the Boundary-Scan chain.

Testing is simplified if in-circuit, bed-of-nails testing can be used to supplement Boundary-Scan (2). In particular, this can help *locate* as well as detect faults, which is a great aid to the manufacturer in diagnosing the causes of PCB failure. This is especially true for PCBs which are only partially populated with Boundary-Scan ICs. Boundary-Scan allows the manufacturer to obtain a higher fault coverage that exceeds that from in-circuit testing. It can provide virtually 100% fault coverage for opens and stuck-at pins, while providing a high coverage of shorts (bridging faults). A potential problem, as with all scan methods, is the time required to scan in and scan out long strings of test values.

Examining recent, representative applications of Boundary-Scan illustrates many of the practical issues associated with use of this technology. Matsushita Electric Industries needed more cost-effective manufacturing and test methods in the face of trends toward significantly more digital components, reduced nodal access on PCBs, and an increasingly competitive marketplace for its diversity of consumer and professional electronics products (6). Their strategy was to obtain 100% fault coverage as early as possible in the manufacturing process. This required tests for opens and shorts on all testable solder joints, checking the presence and basic function of all components, and checking component pins for damage due to electrostatic discharge. Their assembly yield was about 85%. In-circuit test detected around 70% of the faults, resulting in 95% yield after in-circuit test and repair. The remaining faults were not found until a "hot mock-up" test or after final assembly. The increased cost of low, early fault coverage can be seen in the 5 minute average test and repair time for the in-circuit test, compared with 30 minutes for the "hot mock-up" test.

In a joint program with Hewlett-Packard (HP), boards were selected for analyzing and re-engineering manufacturing and test procedures. One board contained 1377 nodes, 5299 solder joints of which 4593 were testable, 450 discrete analog components, and 96 digital ICs, including 9 ASICS (Application Specific ICs) with Boundary-Scan. A set of seven test types was able to achieve 100% fault coverage, even though no *single* test type was able to achieve 100%. The tests were mixed, including in-circuit probe access as well as Boundary-Scan. Notable benefits achieved were the elimination of "hot mock-up" tests, the ability to distinguish between design faults and manufacturing defects early in the product lifecycle, a 15% reduction in probe count for in-circuit testing, a one-third reduction in test development time, and a three to four week reduction in board design time. Boundary-Scan reduced the need for in-circuit test points while improving fault coverage. 100% fault coverage guaranteed that incorrectly functioning boards were the result of *design* errors. Boundary-Scan reduced the time to write ASIC tests because the test process is more automatic. It also reduced design and fixturing time with only a 1 to 5% increase in IC costs to incorporate Boundary-Scan.

ISSUES IN APPLYING BOUNDARY-SCAN

Motorola has reported on the implementation of Boundary-Scan in the PowerPC RISC microprocessor family (7). The PowerPC 602, 603e, and 604 contain 1.6, 2.6, and 3.6 million transistors and are designed for consumer electronics, portable, and desktop PCs, respectively. Motorola implemented Boundary-Scan in a customized rather than a standard fashion. Their PowerPC implementation of Boundary-Scan (IEEE 1149.1) was also used for internal chip testing. This included IBM's Level Sensitive Scan Design (LSSD) internal test methodology, embedded RAM BIST, IDDQ (quiescent CMOS power supply current), emulation, and debug support. Their report focuses on troublesome implementation issues rather than typical design practice.

Originally, LSSD-based testing was not fully compliant with the 1149.1 standard. In particular, unless the control signals for LSSD testing exercised superiority over the 1149.1 boundary register, as discussed in Appendix A of the 1149.1 standard, it was difficult to maintain LSSD rules. This meant that several of the LSSD controlling pins were not Boundary-Scan testable, thereby violating compliance (8). This problem was solved by revision 1149.1a-1993 which introduced the "compliance enable" concept. Private instructions rather than RUNBIST were used to execute an embedded RAM BIST because the control structures for initialization were not compatible. Private instructions were also used for IDDQ testing, emulation, and debug support. They note that in PowerPC designs, most implementation difficulties were related to implementing and verifying SAMPLE rather than EXTEST (5). This was a surprise because the opposite had been believed during the development of the 1149.1 standard.

Designers generally divide device operation into *normal system* and *test* modes and are aware that SAMPLE must not interfere with normal system operation. However, there is a third mode, *non-normal system* operation, which includes factory test, debug, power management, and system hard reset. Insufficient consideration of non-normal system operation led to compliance difficulties with the 1149.1 standard.

For example, they note that the PowerPC 603e includes a system hard reset pin (HRESET) and a checkstop input pin (CKSTP). When HRESET is asserted, the IC is reset, and all output drivers are placed in a high impedance state. CKSTP places all the output drivers in a high impedance state, disables most input receivers, and suspends instruction execution. Directly connecting both HRESET and CKSTP to the driver control logic presented several difficulties when implementing the SAMPLE instruction. One difficulty was that the output enable control cells failed to capture the effect of HRESET or CKSTP on the output drivers. The solution was to disable the output driver through an output control cell by adding CKSTP and HRESET as inputs.

Careful design was also required to ensure 1149.1 compliance when providing input isolation during power down operations. SAMPLE captured the state of the Update latch rather than that of the input pin. The solution was to disable the Update latch during SAMPLE, forcing the receiver multiplexer to select the input pin.

Three output pins were initially incorrectly classified as *two-state* outputs because the signals were released to high impedance only during system reset and checkstop conditions. They were reclassified as *three-state* outputs, and an additional output enable control cell was added for these output drivers. They observed that other groups (9) have noted incorrect classification, suggesting that this is a common problem.

Motorola found that compliance-enable pins and private instructions were valuable tools with which to support the sometimes conflicting demands of chip-level and board-level test. Although a considerable amount of internal testing is incorporated in the PowerPC family, INTEST was not supported. One reason was that the cost of documentation and consumer support for INTEST appeared to be very high. Since both *private* and *public* test domains are useful, they recommend that the 1149.1 standard should be expanded to define what constitutes reentry into the public domain from the private domain.

Recently, Tegethoff and Parker from Hewlett-Packard (10) reviewed the current status of the 1149.1 standard. They observed that this standard had become the methodology of choice for discovering manufacturing defects and enabling functional tests. Having long used Boundary-Scan techniques, their company has converted their internal Boundary-Scan protocol to the 1149.1 standard. They present the perspectives of a user of the standard and a developer of test software supporting the standard.

They note that one problem with verification is fixing responsibility for verifying compliance in an implementation. Test experts typically lack the VLSI (very large scale integration) design skills to evaluate the implementation, while design experts have not mastered enough of the subtleties of 1149.1 to assure compliance. They find that most problems associated with 1149.1 in the board test environment come from design errors, which are the result of misinterpretation or involuntary violation of the standard. Parker (8) notes that for several reasons the verification of compliance is very difficult, and a guarantee is virtually impossible. Some reasons he gives are the lack of formalized rules or procedures for checking compliance and the openness of the 1149.1 standard to user-defined extensions of arbitrary complexity. The domain of possible violations becomes infinite when implementers attempt to map a written-in-English standard to electronic circuits.

Since IC designers rely heavily on simulation for design verification, the Boundary-Scan Description Language (BSDL) (8) can be a great help in insuring compliance. BSDL allows the testability features of components complying with the 1149.1 standard to be described in software-readable language. Writing BSDL can uncover compliance errors in the implementation of Boundary-Scan circuits. For example, BSDL cannot describe an illegal configuration in which system logic is placed between boundary register cells and the I/O pins. Programs with different levels of sophistication can be written to check compliance. BSDL was developed as IEEE Standard 1149.1b-1994. Parker (11) notes that, in retrospect, if the development of BSDL had paralleled the development of 1149.1, many of the ambiguities that led to a rewrite of the 1149.1 standard in 1993 might have been avoided.

BSDL specifies those parameters which are *unique* to a particular Boundary-Scan implementation; those elements of a design which are *mandated* by 1149.1 are not included in BSDL descriptions. For example, neither the bypass register nor the TAP state diagram are described. To assure widespread use, BSDL has been implemented as a *subset with standard practices* in VHDL, the VHSIC (very high speed integrated circuit) Hardware Description Language. As Tegethoff and Parker note (10), simulation of 1149.1 features with verification vectors from automatic test pattern generators (ATPG) can ensure that the BSDL description is correct and matches the implementation in silicon.

ATPG simulation success is a de facto test for compliance and functionality. However, since full compliance is impossible to verify, they believe more robust test vectors are required. For example, robust test vectors are used to check the timing relationships between TAP signals. In particular, Tegethoff and Parker (10) consider the timing relation between TCK and TDI. ATPG will most likely have TDI and TCK changing on the same vector to minimize the total number of vectors. All vectors change on the falling edge of TCK, according to the standard, but the proper timing is not checked. They suggest that a subset of the vectors be generated in which TDI and TMS change a vector before, at, and after the falling edge of TCK to check TDI and TCK timing. This will verify that the TAP samples TDI at the correct TCK edge.

They also raise the question of acceptable violations of the 1149.1 standard. They note that while test experts will demand full compliance, design experts can often make a case for a tolerable violation. Of course, chips sold as 1149.1 compliant should be fully compliant. However, they regard violations in internally-used ICs tolerable if they are transparent to any tester or ATPG tool used subsequently and if they will not hinder use of the standard in the board and system test strategy. As an example, they consider a glue logic chip designed in gate array technology for a board with an 1149.1-based test strategy. If designers cannot fit the gates necessary to implement the Boundary-Scan Register (BSR) in the gate array, they might use external, discrete gates to form the BSR cells. An alternative, which might fit all BSR cells on the chip, is to treat bidirectional I/Os as input-only. Although this violates the standard, the loss of coverage on the path from the driver to the bond pad would be an acceptable risk.

Like all standards documents, 1149.1 is the consensus of a committee; revisions and supplements should be expected in light of experience. For example, Tegethoff and Parker (10) discuss subtleties in implementing the SAMPLE instruction. They suggest that the 1149.1 working group should consider demoting SAMPLE to an optional instruction, since it is tricky to implement realistically and may require inordinate overhead for its support. This requires that SAMPLE is first divorced from PRELOAD and that PRELOAD continues to be mandatory.

They also observe that of all the requirements in the 1149.1, *reset* has become a bone of contention for the design community. Implementing reset on a Boundary-Scan chip requires consideration of the reset strategy for the entire system. It is natural to tie TAP resets to the board power-up reset. However, if the board tester or one of the chips asserts reset during a Boundary-Scan test, the TAPs will also be reset. Thus, *two* board resets are required to ensure no surprises during turn-on. They suggest that one might want to modify the 1149.1 standard to make the power-up reset requirement optional.

Parker (8) distinguishes *non-invasive* and *pin-permission* modes of operation in the 1149.1 standard. The standard specifies a set of Boundary-Scan resources which are guaranteed to be independent of the IC's internal logic. In non-invasive mode, the Boundary-Scan resources communicate asynchronously with external circuits to set up tests or read out results. These activities are invisible to normal IC operation. In pin-permission mode, the I/O pins of the IC are Boundary-Scan controlled, effectively disconnecting the IC's internal logic from external circuits.

This distinction is important when power is applied to a board or system to "bring it to life." Applying power must bring the system to an initial state which is a stable starting point for future behavior. All Boundary-Scan ICs must "wake up" in non-invasive mode. When any one of the Boundary-Scan ICs switches to pin-permission mode, this constitutes "radical surgery," and great care may be needed in "post-operative recovery." Parker (8) calls this the "lobotomy problem." What should an IC do when "waking up" from pin-permission mode to assure that the system doesn't enter an unsafe state? Tegethoff and Parker (10) note that the lobotomy problem is as important and potentially more dangerous to the system than the reset problem. To avoid this problem, Hewlett-Packard devised the drive inhibit (DRV_ENH) instruction to put the IC in a safe state with no possible driver fights and the drive enable (DRV_EN) instruction which restores the IC to normal bus operations (12).

EXTENSIONS TO THE 1149.1 BOUNDARY-SCAN STANDARD

In addition to possible, desirable modifications of the 1149.1 standard as discussed above, there is a related 1149.X family of standards (11) which extend the Boundary-Scan concept. The P prefix denotes a standard which has not completed the acceptance procedure. The 1149.1 standard has been criticized as a CMOS (complementary metal oxide semiconductor) compatible standard which is too burdensome for other processes (10). The P1149.2 standard attempts to address this and provide support for performance/delay testing. P1149.4 allows testability of analog pins and passive components connected to digital pins. P1149.5 provides a protocol to facilitate operation of a module test and maintaince bus. An important question for the usefulness of all these standards is whether or not they address the needs of a significant constituency in the electronics industry (10).

The intent of Standard P1149.2 is to support Boundary-Scan for board-level testing as well as internal scan with minimum mandatory features (13). P1149.2 would offer features similar to those provided by 1149.1, but with two significant differences. First, the P1149.2 BSR cells can be shared with the core logic of the component and are not required to have separate serial-shift and parallel-update stages. Second, P1149.2 uses a direct, parallel-access method to enable the different test modes. To allow the use of software tools developed to support 1149.1, a version of P1149.2, P1149.2T, has been proposed which is backward compatible with 1149.1 but provides additional functions. However, it is likely that P1149.2 may not become a standard.

The IEEE P1149.4 working group is defining a mixed-signal test bus that addresses issues of analog testability (14). Desirable goals are the ability to stimulate any analog pin with an analog source and to monitor any analog pin with an analog measurement instrument. A core disconnect capability is required to turn off the pins' core drivers and, perhaps, guard the core from tester-driven stimuli. Since many measurement scenarios require a guard point through which to route currents, the proposed P1149.4 architecture includes a switch to ground. P1149.4 is an extension of 1149.1 and depends on the existence of four or five 1149.1 TAP interface pins and support circuitry.

The P1149.5 standard is for a module test and maintenance (MTM) bus whose *protocol* standardizes a method of communicating test and maintenance commands and serial data between a subsystem test control module (bus master) and the other (slave) modules on the bus (15). The MTM bus supports module test, subsystem test, and subsystem diagnosis using observability and controllability techniques such as scan and Boundary-Scan. It extends a standard test and maintenance protocol developed in the U.S. Department of Defense's VHSIC program. The MTM bus has a multidrop topology which supports these applications and allows one to remove a board without breaking the communications link between other modules on the backplane.

BIBLIOGRAPHY

1. IEEE/ANSI Standard 1149.1a, *A Standard Test Access Port and Boundary Scan Architecture,* New York: IEEE Standards Board, 1993.
2. B. Geuskens and K. Rose, Implementing a CMOS boundary-scan architecture, *Proc. IEEE Int. ASIC Conf.,* 1994, pp. 392–399.
3. L. Y. Ungar, BIT in a boundary-scan environment, *EE-Evaluation Engineering,* **29** (11): 141–149, 1990.
4. GenRad, Inc., *Meeting the Challenge of Boundary Scan,* Concord, MA: GenRad, Inc., 1991.
5. N. Jarwala and C. W. Yau, A New Framework for Analyzing Test Generation and Diagnosis Algorithms for Wiring Interconnects, *Proc. IEEE Int. Test Conf.,* 1989, pp. 63–70.
6. P. Milo, Success with boundary scan, *EE-Evaluation Engineering,* **34** (2): 72–75, 1995.

7. C. Pyron and W. C. Bruce, Implementing 1149.1 in the Power PC™ RISC microprocessor family, *Proc. IEEE Int. Test Conf.,* 1995, pp. 844–850.
8. K. P. Parker, *The Boundary-Scan Handbook,* Boston: Kluwer Academic Publishers, 1992.
9. W. C. Bruce, M. G. Gallup, G. Giles, and T. Munns, Implementing 1149.1 on CMOS microprocessors, *Proc. IEEE Int. Test Conf.,* 1991, pp. 879–885.
10. M. V. Tegethoff and K. P. Parker, IEEE Std 1149.1: Where are we? Where from here?, *IEEE Design & Test of Computers,* **12** (2): 53–59, 1995.
11. K. P. Parker, Observations on the 1194.X family of standards, *Proc. IEEE Int. Test Conf.,* 1994, p. 1023.
12. D. D. Josephson, D. J. Dixon, and B. J. Arnold, Test features of the HP PA7100LC processor, *Proc. IEEE Int. Test Conf.,* 1993, pp. 764–772.
13. B. Dervisoglu, IEEE P1149.2 description and status report, *IEEE Design & Test of Computers,* **9** (3): 79–81, 1992.
14. A. Cron, P1149.4 mixed-signal test bus, *IEEE Design & Test of Computers,* **13** (3): 98–101, 1996.
15. P. Mc Hugh, IEEE P1149.5 module test and maintenance bus, *IEEE Design & Test of Computers,* **9** (4): 62–65, 1992.

BIBICHE GEUSKENS
KENNETH ROSE
Rensselaer Polytechnic Institute