A Novel Delay Fault Testing Methodology Using Low-Overhead Built-In Delay Sensor

Swaroop Ghosh, Swarup Bhunia, *Member, IEEE*, Arijit Raychowdhury, *Member, IEEE*, and Kaushik Roy, *Fellow, IEEE*

***Abstract*—A novel integrated approach for delay-fault testing in external (automatic-test-equipment-based) and test-per-scan built-in self-test (BIST) using on-die delay sensing and test point insertion is proposed. A robust, low-overhead, and process-tolerant on-chip delay-sensing circuit is designed for this purpose. An algorithm is also developed to judiciously insert delay-sensor circuits at the internal nodes of logic blocks for improving delay-fault coverage with little or no impact on the critical-path delay. The proposed delay-fault testing approach is verified for transition- and segment-delay-fault models. Experimental results for external testing (BIST) show up to 31% (30%) improvement in fault coverage and up to 67.5% (85.5%) reduction in test length for transition faults. An increase in the number of robustly detectable critical-path segments of up to 54% and a reduction in test length for the segment-delay-fault model of up to 76% were also observed. The delay and area overhead due to insertion of the delay-sensing hardware have been limited to 2% and 4%, respectively.**

***Index Terms*—Built-in delay sensor (BIDS), built-in self-test (BIST), delay fault testing, segment delay fault, test point insertion, transition delay fault.**

I. Introduction

SHRINKING transistor geometries have resulted not only in an increased frequency of operation but also in an increased process parameter variation. This has posed serious challenges to traditional testing methodologies [1]. At-speed transition-delay-fault testing has become a necessity for high-performance circuits. Scan architectures provide an efficient way to test for delay faults with reasonably high fault coverage. Both external [automatic test equipment (ATE)-based] and built-in self-test (BIST) architectures are popularly used [2]. However, both of these approaches have their own limitations when applied to delay-fault testing. For example, ATE may suffer from: 1) longer test time and storage requirement because it may involve application of two patterns and 2) high cost. A BIST, on the other hand, offers a simple and relatively low cost solution for delay testing. However, BIST circuitry also suffers from limitations: 1) Weighted random patterns applied in BIST usually require a longer test length than the deterministic case for similar coverage. 2) An increase in test time translates to a corresponding increase in energy dissipation during testing, affecting battery life mainly in handheld devices that employ periodic self-test.

Therefore, it is important to explore new techniques for test-time reduction in ATE and BIST without losing the coverage. One of the effective solutions is test point insertion (TPI) for improving the controllability and observability of the internal nodes of a circuit [3]. TPI for stuck-at faults in scan-based BIST has been explored extensively [4], [5]. However, very little work has been reported on delay fault coverage improvement using observable TPI. The primary reason is the unavailability of a delay-sensing hardware that can act as an observation point for the intermediate nodes. An analog time-to-voltage converter (TVC) with a resolution of 0.5 ns has been proposed in [6] for applications in high-energy physics experiments. A path delay fault testing methodology using TPI and successive test clock trimming for BIST is presented in [7]. An automatic test pattern generator (ATPG)-based TPI technique for enhancing the delay fault coverage by breaking the shift dependence using dummy flip-flop (FF) and combinational gate insertion is proposed in [8]. In [9], FFs have been used as observation points for scan-based BIST. However, for delay measurement, this technique may require generation of separate delayed clocks, which comes at the cost of considerably large area/power/design overhead.

A delay-sensor logic, on the other hand, is a pragmatic solution to the delay testing problem because it can sense the delay at the observation points and detect gross and distributed delay defects. Several delay-sensor circuits have also been proposed recently [10], [11]. However, implications to delay fault coverage have not been studied.

To the best of our knowledge, there is no previous work that attempts to improve transition- [1] and segment-delay-fault [12] test coverage and/or test-size reduction using on-chip delay-sensing hardware. Here, we present an integrated approach that uses process-tolerant delay-sensing hardware at points where a significant improvement in coverage and test set reduction is attained. We have applied our approach to a set of ISCAS'89 benchmarks for transition and segment delay faults. The results show significant improvements in fault coverage with small delay and area penalty. Note that the proposed methodology can be used in a go/no-go test where the failing chips can either be discarded or tested for the next lower speed bin.

This paper makes several contributions. We propose:

1) a novel low-overhead built-in delay sensor (BIDS) that can detect delay failures at the internal nodes; it is robust with respect to glitch and can be easily calibrated for process variation;

Manuscript received December 23, 2004; revised June 17, 2005 and October 31, 2005. This work was supported in part by GSRC MARCO. This paper was recommended by Associate Editor S. Hellebrand.

S. Ghosh, A. Raychowdhury, and K. Roy are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ghosh3@purdue.edu; araycho@purdue.edu; kaushik@purdue.edu).

S. Bhunia is with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleaveland, OH 44106 USA (e-mail: skb21@case.edu).

Digital Object Identifier 10.1109/TCAD.2006. 882523

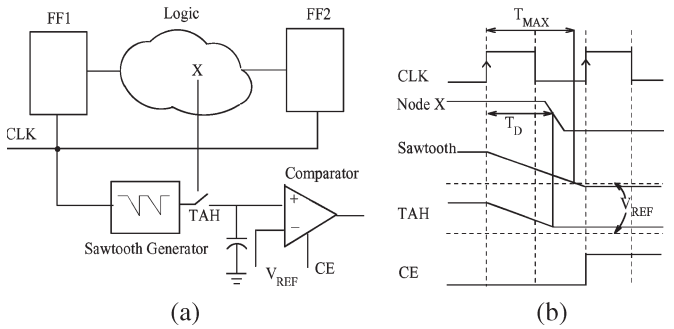

Fig. 1. (a) Delay sensing scheme (X is the node to be probed). (b) Timing diagram of the circuit.

2) a TPI algorithm that can identify the strategic nodes where the BIDS can be inserted to increase the fault coverage or reduce the test application time;
3) a complete transition- and segment-delay-fault test methodology, with capabilities of fault detection and localization, that is compatible with both ATE and BIST architectures.

The rest of this paper is organized as follows: Section II presents the proposed delay sensor and discusses several design issues. Section III presents a complete test architecture and test application procedure using BIDS. Section IV describes a low-complexity probe-point-insertion algorithm. Section V presents and analyzes the results for both external testing and BIST, and Section VI concludes this paper.

II. Delay-Sensor Hardware

A. Principle of Operation

Before going into the detailed description of the delay-sensor hardware, it will be worthwhile to mention the principle of operation of the BIDS. Let us assume a combinational logic block with a state input coming from flip-flop FF1 and the output going to FF2 [Fig. 1(a)]. Let us assume that node X makes a falling transition from "1" to "0" and T_D is the time interval between the rising clock edge and the time when the voltage at node X makes a falling transition [Fig. 1(b)]. Let us assume that the maximum delay that can be tolerated at node X is $T_{\rm MAX}$; hence, $T_D > T_{\rm MAX}$ would mean a delay failure at node X.

We propose the system illustrated in Fig. 1(a) for estimating the delay T_D. A sawtooth generator is so designed that the sawtooth waveform is extracted from the reference clock itself, and it has a pulse duration equal to the time period (T) of the reference clock. The output of the sawtooth generator goes into a track-and-hold (TAH) circuit, and the sampling switch of the TAH is controlled by the observation node X. As long as node X is high, the TAH switch is ON, and the output of the TAH tracks the sawtooth waveform. When X makes a falling transition, it turns the TAH switch OFF, and the output capacitor of the TAH holds its value (say, $V_{\rm TAH}$).

The greater the delay T_D, the lower the $V_{\rm TAH}$. The comparator evaluates at the next clock cycle when the comparator enable (CE) signal goes high [Fig. 1(b)]. The reference voltage $V_{\rm REF}$ of the comparator is a measure of the maximum tolerable

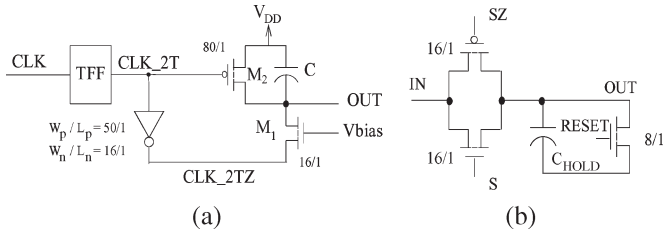

Fig. 2. (a) Schematic diagram of the sawtooth generator. (b) Schematic diagram of the TAH circuit.

delay $T_{\rm MAX}$ of node X. The details of its derivation and implementation will be discussed subsequently. For the observation node X to meet its predefined delay criteria, we require

$$V_{\rm REF} <= V_{\rm TAH}. \tag{1}$$

When (1) is true, the comparator output becomes "1." Note that although we have discussed the case when X makes a falling transition, the same circuit can be used when X makes a rising transition with minor modifications (as discussed in Section III).

B. Design of the Individual BIDS Blocks

Sawtooth Generator: The sawtooth generator [Fig. 2(a)] is based on the principle of constant current discharge. A T flip-flop is used to generate a clock CLK_2T with a period equal to twice the period of the reference system clock. Consider that the node OUT in Fig. 2(a) is precharged to $V_{\rm DD}$ when CLK_2T is low. When CLK_2T goes high (CLK_2TZ goes low), the constant supply voltage $V_{\rm DD}$ provides a constant current through the n-channel MOS (NMOS) M_1. This current discharges capacitor C linearly as long as M_1 is in saturation. During this phase, the p-channel MOS (PMOS) M_2 remains OFF, and the output node shows a linearly falling waveform. At the end of the clock period, the CLK_2TZ signal goes high. This creates a low resistive path across the capacitor through M_2 and thus helps to charge OUT back to $V_{\rm DD}$.

The gate voltage $V_{\rm bias}$ of M1 provides the required current in the saturation region. The discharge is linear (ignoring early effect [13]) as long as M1 is in the saturation region. Hence, we require $V_{\rm ds} >= V_{\rm bias} - V_t$. To ensure this, the output node is allowed to discharge until $V_{\rm OL} = V_{\rm bias} - V_t$ (chosen to be 250 mV, in this case) in a single clock period. To adjust $V_{\rm bias}$, a leakage sensor can be placed close to the bias generator. The output of the leakage sensor will represent the V_t shift of the discharging transistor. Depending on the leakage output of the leakage sensor, $V_{\rm bias}$ can be adjusted to obtain the desired discharging current.

TAH Network: The TAH network for the circuit is a complementary-pass transistor switch with a capacitive load [Fig. 2(b)]. The voltage at observation point X is the input signal S to the TAH. As long as S is high, it will charge capacitor $C_{\rm HOLD}$. To discharge $C_{\rm HOLD}$, an NMOS switch triggered by a RESET signal is used in parallel. The RESET pulse is generated after the comparison between $V_{\rm REF}$ and $V_{\rm TAH}$ has been made. A delayed CE signal can be used as a RESET pulse to discharge $C_{\rm HOLD}$. It is also worth mentioning

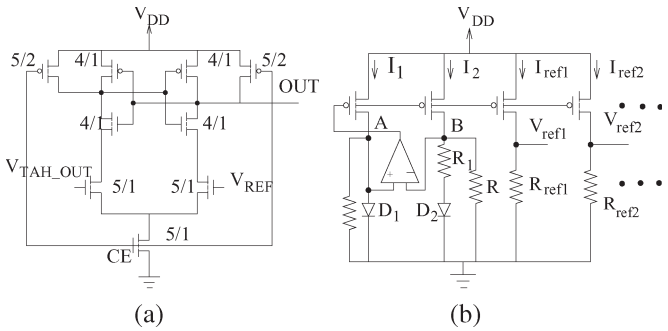

Fig. 3. (a) Schematic diagram of the comparator. (b) Schematic diagram of the reference voltage generator.

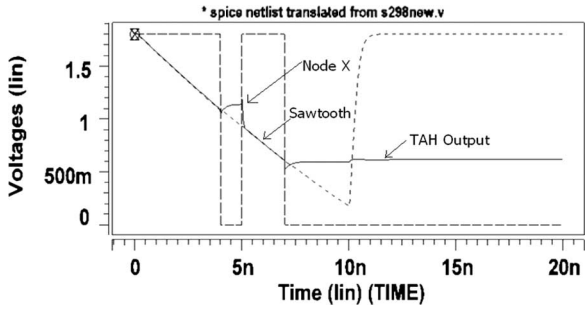

Fig. 4. Spice simulation showing that the proposed BIDS can work even under a glitch in the net to be probed.

that the sampling switch has been made a complementary one to avoid charge injection and clock feed-through [14].

Comparator: The value of the signal at node TAH_OUT goes into the comparator input. The comparator used here is a latch-based sense amplifier, as illustrated in Fig. 3(a). After the TAH phase, CE goes high, and the output of the comparator is noted in the next clock cycle.

Generation of Reference Voltages: The reference voltage $V_{\rm REF}$, which is an input to the comparator stage, must match with the required delay specification. Considering a linearly falling sawtooth waveform, the relationship between the reference voltage and the allowed delay is given by

$$V_{\rm REF} = V_{\rm DD}\left(1 - \frac{T_D}{T}\right) + V_{\rm OL}\frac{T_D}{T} \tag{2}$$

where T is the clock period. In the proposed BIDS, the stable voltage references $V_{\rm REF}$s for the BIDS have been designed [Fig. 3(b)] based on the design of sub-1-V bandgap reference voltage [15]. The operational amplifier equalizes the voltage between nodes A and B. Hence, all the PMOS transistors at the top have the same $V_{\rm gs}$; hence, they mirror the same current, i.e.,

$$I_1 = I_2 = I_{\rm REF1} = I_{\rm REF2} = \cdots. \tag{3}$$

Let V_d be the voltage across diode D1. It has a negative temperature coefficient. dV is the voltage difference between diodes D1 and D2 (of area N times that of D1); hence, $dV = V_T \ln(N)$, where V_T is the thermal voltage that has a positive temperature coefficient. The total current I_2 is given by

$$I_2 = \frac{V_B}{R} + \frac{dV}{R_1} = \frac{V_d}{R} + \frac{dV}{R_1}. \tag{4}$$

Due to the current mirror, the same current is pumped into the reference voltage generator arm. The output reference voltage of the ith arm is thus

$$V_{\rm REF} = R_{\rm REF}\left(\frac{V_d}{R} + \frac{V_T \ln(N)}{R_1}\right). \tag{5}$$

N is chosen such that the net temperature coefficient is zero. Note that any voltage can be generated by changing the value of $R_{\rm REF}$. Several different arms have been shown in the circuit. Furthermore, the output voltage is not dependent on the transistor parameters. Therefore, even under die-to-die parameter variations, the reference generator delivers a stable reference voltage.

It can be mentioned that the bandgap reference usually forms an integral part of all mixed-signal and digital circuits. We can use the bandgap reference already present in the circuit, and we can add the reference generator arms to it to obtain a wide range of temperature-insensitive and stable voltage references. This reference generator can be shared by all the BIDS present in the circuit.

Impact of Process Variation and Input Glitch: It has been mentioned that we generate process-variation-tolerant reference voltages using a modified bandgap reference. The other important BIDS block that can be affected by process variation is the sawtooth generator. Process variation changes the discharge rate of capacitor C and hence, impacts $V_{\rm OL}$ and the choice of $V_{\rm REF}$. To compensate for this, we propose an initial calibration cycle where capacitor C (which is a metal capacitor) is trimmed (using programmable capacitor arrays, PCAs), depending on the process corner. This would ensure a linear discharge from $V_{\rm DD}$ to $V_{\rm OL}$ across all dies. The value of the capacitors in PCA for a chip can be determined in two steps, namely: 1) replicating a large capacitor $C_{\rm nom}$ on the chip and measuring its value $C_{\rm meas}$ using an external resistor (by finding the pole of RC circuit) and 2) scaling the capacitors in PCA by a factor of α, where $\alpha = C_{\rm meas}/C_{\rm nom}$. However, this method may not be accurate because of the presence of a large pin capacitance, and one may consider using laser trimming for better accuracy.

Since most of modern-day processors do contain analog circuits, it can be expected that the process is able to handle the metal capacitors. However, in case of process incompatibility a MOS capacitor can be used instead at the cost of poor linearity of the circuit. The TAH circuit is a transmission gate with large-sized transistors. Hence, process variation cannot considerably impact the functionality or performance of this block. Finally, the comparator is differential in nature, and die-to-die variation cannot impact its functionality. Furthermore, latch-based comparators, which are tolerant to within-die variation, have been reported in [16] and studied in the design of proposed BIDS. We have also studied the effect of glitch [13] in the node to be probed. It can be noted that the TAH follows the sawtooth once the glitch dies and holds the value only at the last transition (Fig. 4). Thus, the proposed BIDS is robust with respect to glitches.

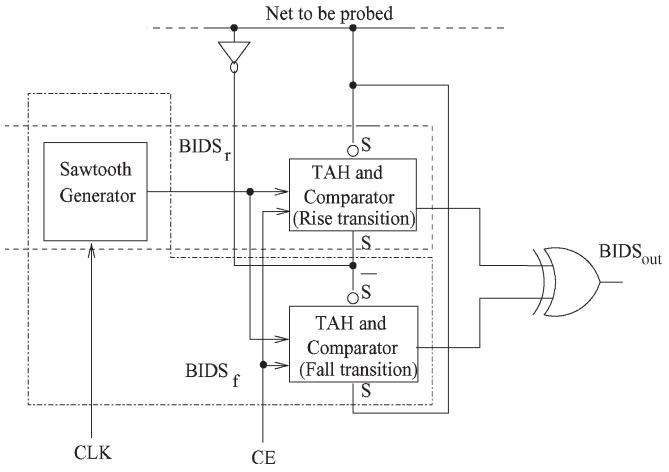

Fig. 5. Schematic diagram for sensing both the rise and fall transition delays at a node using BIDS.

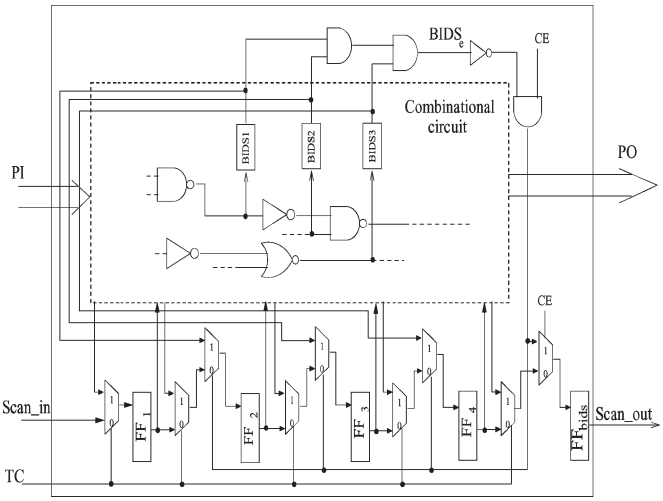

Fig. 6. ATE-based test architecture using delay sensing hardware.

Delay Sensitivity of BIDS: The delay sensitivity of the proposed BIDS is mainly determined by two factors: 1) TAH offset δv_1 and 2) comparator offset δv_2. Therefore, the worst case sensitivity is determined by $\delta v = \delta v_1 + \delta v_2$. The typical value of δv is found to be 50 mV in our case. For a voltage swing of 1.4 V, the delay resolution turns out to be $T/(1.4/0.05) = T/28$, where T is the time period of the clock. The sensitivity can be modified by tuning δv_1 and/or δv_2 using offset compensation techniques in the design of comparator and TAH.

Note that the basic principle of operation of BIDS is very similar to the TVC method proposed in [6]. In TVC, the capacitor charges between the START and STOP pulses; thus, the time interval between START and STOP is converted to the voltage of a capacitor. In BIDS, the delay is first converted to a voltage and then compared against the voltage corresponding to the expected delay. However, the following differences with TVC can be noted: 1) The operation of TVC needs a width generator to convert the START and STOP pulses to appropriate voltages for controlling the charging of capacitor. On the other hand, BIDS does not require generation of control voltages. The discharging of $C_{\rm HOLD}$ starts with the sawtooth and stops with the rising/falling signal at the node to be probed. 2) The minimum measurable time by TVC is 5 ns, whereas it is $T/28$ ns in case of BIDS. The maximum measurable time of BIDS is limited by the clock period. Furthermore, the resolution of BIDS (i.e., T/28) can be much better than TVC resolution, which is reported to be 0.4 ns. 3) The TVC is affected by parasitic capacitances that create transients and degrade the linearity of the capacitor charging. BIDS; on the other hand, it can function correctly even under glitch. 4) The effect of process variation in TVC has been mitigated by following different layout techniques and overdriving the switching transistors. In BIDS, most of the components are independent of process variation by virtue of their design.

III. Transition Delay Testing With BIDS

In the previous section, we showed that BIDS can detect a delay in falling transition. In this section, we present a simple circuitry (Fig. 5) that can detect delay in both rising and falling transitions. We connect BIDS$_r$ and BIDS$_f$ in parallel, which receive the original (S) and complemented $(\bar{S})$ signals from the net under consideration. The decision for a falling transition is made by BIDS$_f$ and that of a rising transition is made by BIDS$_r$. In case the delay meets the given criteria, the output of the corresponding BIDS is "1." Hence, if we XOR the outputs of BIDS$_r$ and BIDS$_f$, then a delay fault (either slow-to-rise or slow-to-fall) is detected if the output of the XOR (BIDS$_{\rm out}$) is "0."

One interesting observation is that we do not need any response analyzer for a pass–fail decision. Instead, the pass–fail decision is automatically available at BIDS$_{\rm out}$. Moreover, insertion of BIDS has minimal area and delay overhead. A single sawtooth generator and reference voltage generator can be used for all BIDS blocks. However, note that crosstalk can degrade the BIDS accuracy if the sawtooth signal is not shielded properly.

The delay overhead due to BIDS insertion is less than 1% of the critical-path delay for s1196 (the maximum among all the ISCAS benchmarks considered here) and less than 0.4% on an average. The area overhead (active area) was computed for the number of probe points beyond which there was no considerable improvement in the fault coverage. For larger benchmarks such as s35932 (s15850), the area overhead is just 0.7% (2.2%), with 20 probe points. However, for small benchmarks such as s838, the area overhead was higher, leading to an average area increase of 3.9%.

A. ATE-Based Test Architecture

An ATE-based test architecture with four BIDSs is depicted in Fig. 6. It is obtained by modifying the conventional scan architecture. The scan chain consists of four flip-flops, namely, FF1, FF2, FF3, and FF4. The test application procedure for skewed-load transition-delay-fault testing [3] using ATE is given as follows: The setup vector v_1 is shifted into the scan chain (using the Scan_in signal) by keeping the test control (TC) and CE signals low and applying the clock. The corresponding primary inputs (PIs) are applied by the ATE once v_1 has been shifted in. After the application of v_1, activation vector

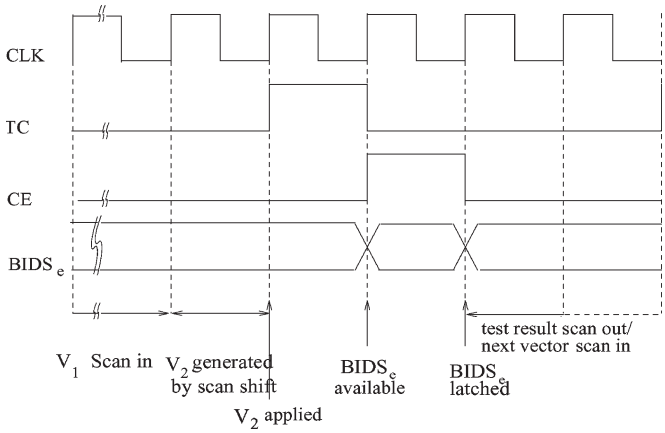

Fig. 7. Timing diagram for the test application in external testing.

v_2 is obtained by a one-bit shift and applied to the circuit under consideration. The corresponding PI vectors are simultaneously applied, and signal TC is asserted. In the following clock cycle, TC is deasserted, and the primary output (PO) is observed by the ATE. The CE signal is asserted for one clock cycle (Fig. 7) so that the BIDSs can verify the delays of the nets under consideration and produce an output signal BIDS_e. The BIDS_e signal is generated by ANDing all the individual BIDS outputs. If the BIDS_e output is low, it means that at least one of the observation nodes has suffered a delay failure.

In the following cycle, the BIDS_e signal is latched in a separate flip-flop FF_{BIDS}, which is introduced to avoid an extra pin for observing the BIDS error signal. BIDS_e is inverted and ANDed with CE, and fed to the select signal of a multiplexer chain. If both BIDS_e and CE are high (i.e., the comparison between V_{REF} and V_{TAH} has been performed, and there is a delay violation), then the BIDS values are latched in a part of the scan chain. Otherwise, the scan chain latches the test responses from the combinational logic. Enhanced-scan and broadside test application [3] for scan-based delay testing can be similarly achieved with the proposed architecture using BIDS. The new setup vector is shifted in, and the scan-chain contents are shifted out simultaneously from this cycle, as depicted in Fig. 7. When the ATE observes an error signal on FF_{BIDS} during Scan_out, it knows that the Scan_out values at the following cycles are BIDS outputs. This can be subsequently used for diagnosis purposes to localize the failure. It can be noted that the BIDS delay verification causes an extra clock cycle delay compared to the normal scan test, which can be negligible, considering the length of scan chain.

B. Modified Test-Per-Scan BIST Architecture

A modified test-per-scan BIST-based test architecture with three BIDSs, namely, BIDS1, BIDS2, and BIDS3, is depicted in Fig. 8. The scan chain consists of four flip-flops, namely, FF1, FF2, FF3, and FF4. The test patterns for the PI and the scan latches are generated as weighted random patterns by a linear feedback shift register (LFSR). The PO and scan latch outputs are provided to a multiple-input signature register (MISR). The signatures generated by MISR are fed to an output response analyzer (ORA). At the end of each test pattern

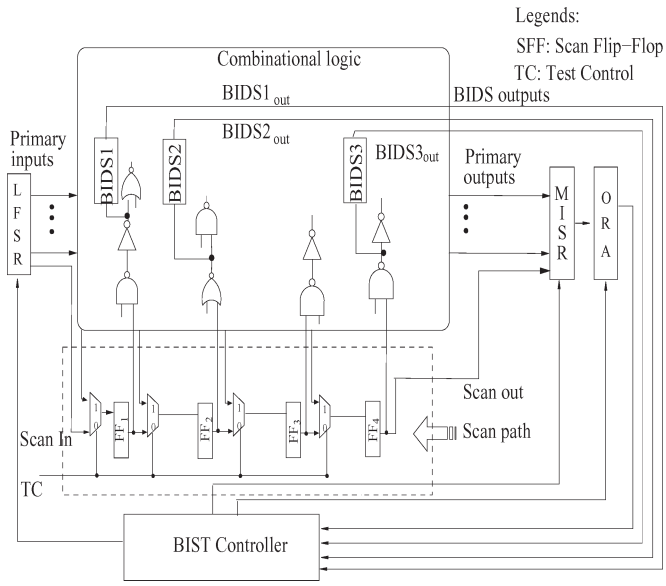

Fig. 8. Test-per-scan BIST architecture using delay sensing hardware.

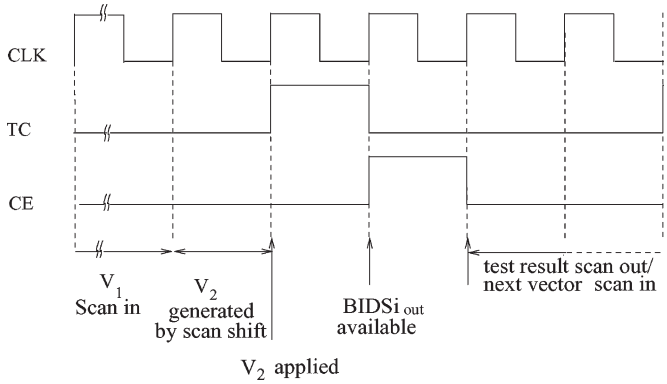

Fig. 9. Timing diagram for the test application in test-per-scan BIST.

application, ORA and BIDS output signals are provided to the BIST controller.

Test application procedure using skewed-load transition-delay-fault testing [3] is given as follows: The setup vector v_1 is shifted into the scan chain (using the Scan_in signal) by keeping the TC and CE signals low and applying the clock. The corresponding PIs are applied by the LFSR once v_1 has been shifted in. After the application of v_1, activation vector v_2 is obtained by a one-bit shift and applied to the circuit under consideration. The corresponding PI vectors are simultaneously applied, and signal TC is asserted. In the following clock cycle, TC is deasserted, and the PO is fed to the MISR. The CE signal is asserted for one clock cycle (Fig. 9) so that the BIDSs can verify the delays of the nets under consideration and produce output signal $\text{BIDSi}_{\text{out}}$ for the ith BIDS. This completes one cycle of test application. Test application for enhanced-scan or broadside test application can be similarly derived.

Note that by using the delay sensor, it is possible to localize the logic violating the delay specification at that node. However, it is not possible to detect the size of the delay failure. This is because the delay sensor only determines the delay violation at that node.

IV. Observation Point Insertion

In this section, we propose an algorithm to choose a set of probe points where the placement of on-chip BIDS can be beneficial. We have considered transition- and segment-delay-fault models for TPI. There are two main objectives to be considered while choosing the probing points: 1) to reduce the number of test vectors and thus the test time and 2) to increase the fault coverage. Note that to satisfy the first objective, we should choose points that are at the intersection of a large number of signal propagation paths and are likely to have delay violations. For the second objective, we should check hard-to-detect faults and insert probing hardware at nodes that are usually hard to observe. In this paper, we have used observability/controllability parameters for each gate (or node) and chosen a node that is: 1) easily controllable but have poor observability and 2) lying at the intersection of many incoming path segments. Our probe metric accounts for both of these factors during probe-point selection.

A. Probe-Point-Insertion Algorithm for Transition Delay Faults

The probe-point-insertion algorithm for transition delay faults begins by creating a graph G of the input netlist with the gates as "nodes" and connections between two gates as "edges." The nodes are sorted topologically. Next, by traversing the graph in topological order, the controllability and observability values of each input and output node are computed in a manner described in [18]. From the 0 and 1 controllability values, we compute the probability of a rising and a falling transition for each edge of graph G. As an example, for a two-input NAND gate

$$C_0(out) = C_1(in1)C_1(in2); \quad C_1(out) = 1 - C_0(out)$$

$$O(in1) = C_1(in2)O(out); \quad O(in2) = C_1(in1)O(out)$$

$$P_r(out) = P_f(in1)C_1(in2)C_1(in2) + P_f(in2)C_1(in1)C_1(in1)$$

$$P_f(out) = P_r(in1)C_1(in2) + P_r(in2)C_1(in1)$$

where $C_0(C_1)$ is the 0(1) controllability, O is the output observability, and $P_r(P_f)$ is the rise (fall) transition probability. For a rise transition (i.e., $0 \rightarrow 1$), the first pattern of both inputs must be kept at "1" to generate a "0" at the output, while the second pattern of one of the inputs should be "0" to generate a "1." On the other hand, for a fall transition (i.e., $1 \rightarrow 0$), the first pattern of one of the inputs can be "don't care," and the second input should be kept at "0," whereas the second pattern of both inputs should be kept at "1." Therefore, the first pattern is not taken into account for computing P_f. The procedure of computing P_r/P_f can also be extended for other logic gates.

At the inception of the algorithm, all path segments are marked to be "uncolored." If a probe point is inserted at a node output, we "color" all path segments in its fan-in cone. Now for each node, we compute the number of uncolored path segments in its fan-in cone. To determine the best node for insertion of probe hardware, we define the probing metric for the ith node as shown at the bottom of the page.

T_i and O_i represent the testability and observability of the ith node's output, respectively, and $w1$ and $w2$ are the weights assigned to the two components of the probe metric. Note that by using testability T_i, we target the nodes that have good controllability but poor observability. A higher value of $(P_r + P_f)$ at the input of a gate indicates that the gate is easily controllable. Similarly, a higher value of $(1 - O_i)$ indicates the poorness of the gate's observability. Therefore, a higher value of T_i indicates that the node has better controllability and poorer observability. An observation point at such points is expected to improve the test length if a large number of faults is incident on these nodes (which is determined by NP_i). Weights $(w1, w2)$ are initialized to (1, 1), but they can be tuned experimentally to maximize the coverage for a given test length. An ideal probe point must have maximum probing metric value. To ensure that probing points are inserted only in paths that meet the slack constraint, we maintain the worst case slack values at all nodes (by performing a static timing analysis, STA). Depending on the timing slack available in a node chosen for BIDS insertion, we have the following options.

1) If the chosen node belongs to a timing-critical path and does not meet the slack constraint, then the insertion is aborted, and the node is marked "colored."
2) If the slack constraint is met, then the node is selected for insertion. At the same time, slacks of all timing paths passing through the selected node are updated. This can be done by simply subtracting the BIDS delay overhead from the slacks of those paths. Note that we need not run a complete timing analysis after each probe-point insertion. Once the path slacks are updated, slacks of all nodes lying on these paths are also modified accordingly. This completes one iteration of the probe-point-insertion algorithm.

$$Pm_i = w1 * T_i + w2 * NP_i$$

$$\text{where} \quad T_i = (1 - O_i) * (P_r(i) + P_f(i))$$

$$NP_i = \text{normalized number of uncolored paths in the fan-in cone of the } i^{th} \text{ node}$$

$$NP_i = \frac{\text{number of uncolored incident paths}}{\text{maximum number of uncolored paths for any edge}}$$

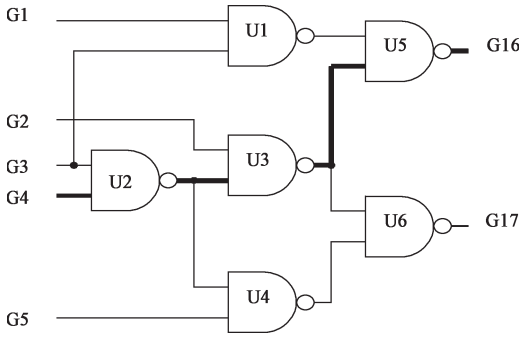

Fig. 10. Timing-critical path in C17.

The operation of the probe-point-insertion algorithm can be explained with the help of a small example circuit (Fig. 10). Let us assume that the number of probing points is two. The algorithm begins with reading the netlist and creating a graph G, and sorting it topologically ($U1{-}U2{-}U3{-}U4{-}U5{-}U6$ in this example). The testabilities, normalized number of paths, and probing metrics for the nodes are also computed. By performing STA, we note that path $G4{-}U2{-}U3{-}U5{-}G16$ is timing critical (Fig. 10). From the computation, it is observed that node $U4$ has the maximum probe metric value (excluding nodes $U5$ and $U6$, which are POs). Since $U4$ does not violate the slack requirement, it is selected for the probe-point insertion. Note that we are not required to update the slacks because $U4$ does not belong to the timing-critical path. Now, the fan-in cone of $U4$ is colored so that the probabilities of selection of all nodes in its fan-in cone are reduced. Additionally, the observability of selected node $U4$ is changed to 1, and the observabilities of all nodes in its fan-in cone are recomputed. The process of finding the next probe point is repeated to insert the next probe hardware into the circuit. The pseudocode of the probe-point-insertion algorithm is presented as follows:

Algorithm InsertProbe ()
Input: Spice netlist, #probe points (Y), slack constraint, target coverage;
Output: A set of probe points;
begin
1. Read the netlist and create a graph (G);
2. Order the nodes of G topologically;
3. Traverse G and compute;$C(0,1)$ and O for each node;
4. Using STA, find worst case timing slack for each gate;
   do
5. Start traversing G from PIs;
6. **For** each node i **do**
7. Find testability T_i;
8. Find number of uncolored path segments (NP_i) to node i;
9. Find probing metric $Pm_i = f(T_i, NP_i)$;
   end for
10. Get node X with $max(Pm_i)$;
11. **If** X meets the slack constraint **then**
    11.1 Mark all path segments in fan-in cone of X as colored;
    11.2 Update slack values of all paths containing node X;
    11.3 Recompute the observabilities of the fan-in cone;
    else
12. Mark node X as colored, and continue;
    while (Y probing points are inserted or the target coverage is met);
end.

The algorithm is computationally efficient with an overall complexity of $O(n^2)$ since it is dominated by topological sorting.

B. Probe-Point Insertion for Segment Delay Faults

The segment-delay-fault model has emerged as a popular delay fault model to test for possible delay violations in path segments [12], [17]. The segment length is an input parameter to the model, which varies from 1 (corresponding to the transition-delay-fault model) to the longest segment length (corresponding to the critical-path length). The probe-point-insertion algorithm for the segment delay fault is identical to the probe-point-insertion algorithm for the transition delay fault. The probing metric, however, is different in this case. The probing metric for segment delay fault for segment length L shown at the bottom of the next page, where, as before, T_i is the testability of the ith node's output edge and $w1$ and $w2$ are the respective weights assigned to each component of the probe metric.

V. Results

In the simulation setup, the ISCAS'89 benchmark circuits are mapped to the Leda standard-cell library, and simulations are performed with Taiwan Semiconductor Manufacturing Company (TSMC) 180-nm technology. Once the nodes for probe insertion are determined, the nodes so obtained are treated as pseudo-POs. The test coverage and the test patterns for the transition delay fault are obtained using the Synopsys Tetramax ATPG tool [19]. The area overhead is limited by setting the maximum number of test points to 20, and the tolerance in critical-path delay overhead is set to 2%.

Transition Faults: Table I shows the fault coverage for the original and modified circuits (after TPI) for a set of 50, 100, 150, and 250 test patterns generated by Synopsys Tetramax ATPG tool [19]. Note that the test patterns are generated for enhanced-scanlike arbitrary two-pattern test application. It can be observed from Table I that a maximum of 31.27% coverage improvement is attained for benchmark s1196 with only 50 patterns. On average, the coverage improvement was found to be 15.15% for 50 test patterns. Table II presents the test-length reduction for the target coverages of 80%, 85%, 90%, and 95%. A reduction in test length of up to 67.5% is observed for circuit s9234 (for 95% of the target coverage). It can also be observed from Table II that an average reduction in test length of 48.47% is attained for a target coverage of 80%.

Table III shows the fault coverage for the original and modified circuits (after TPI) for a set of 50, 100, 150, and 250 random test patterns generated by Synopsys Tetramax ATPG tool [19]. Note that the test patterns are generated for

TABLE I
FAULT COVERAGE IMPROVEMENT (%) WITH A FIXED NUMBER OF TEST PATTERNS FOR ATE-BASED TESTING

Ckt	50 Patterns			100 patterns			150 patterns			250 patterns		
	ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp
s898	65.91	78.36	18.88	81.17	88.81	9.41	91.27	96.29	5.50	100	100	0
s1196	61.91	81.27	31.27	78.35	91.58	16.88	86.89	95.77	10.21	95.5	98.78	3.43
s1423	83.47	91.26	9.33	92.58	96.76	4.51	97.69	99.24	1.58	100	100	0
s5378	71.55	83.09	16.12	80.84	89.73	10.99	87.35	92.01	5.33	92.6	93.47	0.93
s9234	60.88	70.79	16.27	73.09	85.41	16.85	79.21	89.08	12.46	86.96	95.65	9.99
s13207	70.7	78.83	11.49	76.75	84.96	10.69	81.49	88.43	8.51	88.36	94.96	7.46
s15850	70.47	79.3	12.53	78.31	86.37	10.29	83.73	91.17	8.88	87.97	93.93	6.77
s35932	94.45	99.47	5.24	99.47	100	0.53	99.98	100	0.02	100	100	0
Avg.			15.15			10.01			6.56			3.57

TABLE II
TEST-TIME (NUMBER OF TEST PATTERNS) IMPROVEMENT FOR A FIXED TARGET TEST COVERAGE FOR ATE-BASED TESTING

Ckt	Target coverages											
	80%			85%			90%			95%		
	ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp
s898	122	79	35.24	124	80	35.48	151	106	29.80	213	183	14.08
s1196	125	61	51.20	154	76	50.64	188	93	50.53	248	141	43.14
s1423	59	36	38.98	62	40	35.48	92	53	42.39	124	63	49.19
s5378	93	35	62.36	154	63	59.09	214	110	48.59	304	300	1.31
s9234	159	78	50.94	223	109	51.12	317	170	46.37	475	154	67.57
s13207	128	71	44.53	191	92	51.83	319	161	49.52	511	259	49.31
s15850	127	54	57.48	192	88	54.16	320	160	50.00	509	268	47.34
s35932	51	27	47.05	57	30	47.36	59	31	47.45	64	32	50.00
Avg.			48.47			48.14			45.58			40.24

TABLE III
FAULT COVERAGE IMPROVEMENT (%) WITH A FIXED NUMBER OF PATTERNS FOR TEST-PER-SCAN BIST

Ckt	50 Patterns			100 patterns			150 patterns			250 patterns		
	ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp
s898	55.82	57.98	3.86	53.61	55.57	3.65	53.16	55.67	4.72	53.66	54.17	0.95
s1196	57.66	75.15	30.33	72.26	83.97	16.20	76.46	85.16	11.37	85.34	92.67	8.58
s1423	84.74	92.41	9.05	90.44	95.91	6.04	94.42	97.88	3.66	97.72	99.55	1.87
s5378	70.53	77.97	10.54	78.97	85.97	8.86	84.89	89.21	5.08	90.21	92.44	2.47
s9234	56.69	65.18	14.97	63.35	70.68	11.57	64.71	71.13	9.92	75.07	81.66	8.77
s13207	64.66	69.16	6.95	68.42	73.21	7.00	72.94	76.59	5.00	76.97	80.56	4.66
s15850	70.90	75.41	6.36	73.31	77.73	6.02	77.37	81.30	5.07	82.87	86.20	4.01
s35932	97.80	99.50	1.73	99.72	99.97	0.25	99.94	100	0.06	100	100	0
Avg.			10.47			7.44			5.61			3.92

enhanced-scanlike arbitrary two-pattern test application. It can be observed from Table III that a maximum of 30.3% coverage improvement is attained for benchmark s1196 with only 50 patterns. On average, the coverage improvement was found to be 10.47% for 50 test patterns. Therefore, the test confidence is boosted significantly for the same test time. Table IV presents the test-length reduction for target coverages of 80%, 85%, 90%, and 95%. A reduction in test length of up to 85.4% is observed for circuit s1196 (for 90% of the target coverage). It can also be observed from Table IV that an average reduction in test length of 48.59% is attained for a target coverage of 90%. Note that in Tables II, and IV we could not obtain the exact target coverage (as desired) of the original circuit due to the ATPG tool limitation. Therefore, we obtained the test length of modified circuit by equalizing its test coverage to that of the original circuit.

We also observed the impact of number of probe points on the transition fault coverage. The result indicated that a

---

$$Pm_i = w1 * T_i + w2 * NS_i$$

where $T_i = (1 - O_i) * (P_r(i) + P_f(i))$

$NS_i =$ normalized number of uncolored critical segments of length L in the fan-in cone of the ith node

$$NS_i = \frac{\text{number of uncolored critical incident paths}}{\text{maximum number of uncolored critical paths for any edge}}$$

TABLE IV
TEST-TIME (NUMBER OF TEST PATTERNS) IMPROVEMENT FOR A FIXED TARGET TEST COVERAGE FOR TEST-PER-SCAN BIST

Ckt	Target coverages											
	80%			85%			90%			95%		
	ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp
s898	39	34	12.82	57	50	12.28	62	51	17.74	58	55	5.17
s1196	271	174	35.79	258	98	62.01	255	37	85.49	263	68	74.14
s1423	66	44	33.33	100	60	40.00	134	59	55.97	177	145	18.07
s5378	134	25	81.34	221	148	33.03	270	78	71.11	251	53	78.88
s9234	166	111	33.13	166	63	62.04	166	81	51.20	177	80	54.80
s13207	176	49	72.15	369	258	30.08	367	214	41.68	351	216	38.46
s15850	199	151	24.12	442	284	35.74	423	285	32.62	463	293	36.71
s35932	83	53	36.14	87	56	35.63	91	61	32.96	93	65	30.01
Avg.			41.10			38.85			48.59			42.04

TABLE V
PERCENTAGE INCREASE IN THE NUMBER OF DETECTED FAULTS AND REDUCTION IN TEST LENGTH FOR DIFFERENT SEGMENT LENGTHS L

Ckt	L	Test coverage improvement									Test length improvement					
		Non-robust			Robust			Combined			Non-robust		Robust		Combined	
		ORG	TPI	%imp	ORG	TPI	%imp	ORG	TPI	%imp	#pat	%imp	#pat	%imp	#pat	%imp
s838	2	33	36	9.09	33	36	9.09	38	41	7.89	575	42.5	575	42.5	575	42.5
	3	26	28	7.69	26	28	7.69	30	32	6.66	600	40.0	600	40.0	600	40.0
	5	19	20	5.26	20	21	5.00	22	23	4.54	575	42.5	575	42.5	575	42.5
s1196	2	422	427	1.18	376	383	1.86	462	467	1.08	980	2.0	465	53.5	980	2.0
	3	394	401	1.77	369	379	2.71	455	462	1.53	980	2.0	510	49.0	980	2.0
	5	339	357	5.30	279	293	5.01	377	394	4.50	900	10.0	515	48.5	900	10.0
s1423	2	216	218	0.92	195	202	3.58	218	220	0.91	830	17.0	880	12.0	830	17.0
	3	216	220	1.85	189	197	4.23	220	224	1.81	240	76.0	370	63.0	265	73.5
	5	221	229	3.61	165	223	35.15	223	231	3.58	555	44.5	370	63.0	555	44.5
s5378	2	308	309	0.32	274	303	10.58	325	327	0.61	935	6.5	300	70.0	935	6.5
	3	361	363	0.55	300	346	15.33	385	390	1.29	935	6.5	270	73.0	840	16.0
	5	381	383	0.52	305	366	20.00	410	433	5.60	935	6.5	460	54.0	470	53.0
s9234	2	199	201	1.00	118	150	27.11	204	206	0.98	860	14.0	465	53.5	860	14.0
	3	205	207	0.97	105	147	40.00	213	223	4.69	860	14.0	465	53.5	860	14.0
	5	189	189	0	53	82	54.71	195	201	3.07	950	5.0	465	53.5	860	14.0
s13207	2	148	170	14.86	74	96	29.72	150	173	15.33	370	63.0	410	59.0	370	63.0
	3	155	179	15.48	71	89	25.35	157	182	15.92	410	59.0	505	49.5	400	60.0
	5	132	160	21.21	39	48	23.07	133	162	21.80	410	59.0	550	45.0	410	59.0
s15850	2	187	193	3.20	137	168	22.62	190	196	3.15	730	27.0	375	62.5	730	27.0
	3	197	203	3.04	141	167	18.43	199	205	3.01	900	10.0	400	60.0	900	10.0
	5	159	170	6.91	95	110	15.78	163	175	7.36	710	29.0	605	39.5	710	29.0
s35932	2	361	418	15.78	327	381	16.51	389	428	10.02	643	35.7	532	46.8	643	35.7
	3	340	389	14.41	301	361	19.93	371	401	8.08	695	30.5	586	41.4	695	30.5
	5	329	370	12.46	291	337	15.80	359	388	8.07	724	27.6	590	41.0	724	27.6
Avg.				6.14			17.88			5.89		27.90		50.67		30.55

maximum of 10 probe points are necessary to achieve significant improvement in test coverage. The same study for all the benchmarks indicated an identical behavior to conclude that 8–20 probe points are sufficient for the purpose.

Segment Delay Faults: The segment-delay-fault simulator SIGMA [17] is implemented using the C programming language. The segment length L has been chosen to be 2, 3, and 5, and a set of 1000 random input vectors is applied. Due to the robust fault propagation requirement, the fault coverage of segment delay faults is lower than that of transition delay faults [17]. Hence, the metric of interest in this paper is the percentage increase in the number of detected critical faults when the same set of input vectors is applied to the circuit both before and after the test points are inserted. Table V depicts the number of detected critical-path segments before and after TPI and improvement (in percentage). It can be noted that a maximum of 21.21%, 54.71%, and 21.8% improvement is achieved for nonrobust, robust, and their combination, respectively. In Table V, we have depicted the number of test patterns (#pat) required to detect a similar number of critical-path segments (as original) after TPI. We observe that test time can be reduced significantly (as high as 76%, 73%, and 73.5% for nonrobust, robust, and their combination, respectively) for a similar test confidence.

VI. CONCLUSION

This paper presents a novel delay fault testing methodology using an on-die delay sensor. The design of a low-overhead process-tolerant delay testing hardware has been presented. An algorithm has been proposed to judiciously select probe points in the combinational circuit where the delay sensors should be inserted for maximizing coverage and/or test-time improvement. Simulation results show that the proposed method is promising for delay testing in external (ATE-based) and scan-based BIST.

REFERENCES

[1] A. Krstic and K.-T. Cheng, *Delay Fault Testing For VLSI Circuits*. Boston, MA: Kluwer, 1998.

[2] S. Sunter, "BIST vs. ATE: Need a different vehicle?" *Proc. Int. Test Conf.*, 1998, p. 1148.

[3] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing, for Digital, Memory and Mixed-Signal VLSI Circuits*. Norwell, MA: Kluwer, 2000.

[4] H. C. Tsai, K. T. Cheng, and V. D. Agrawal, "A testability metric for path delay faults and its application," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2000, pp. 593–598.

[5] M. J. Geuzebroek, J. T. V. Linden, and A. J. van de Goor, "Test point insertion that facilitates ATPG in reducing test time and data volume," in *Proc. Int. Test Conf.*, 2002, pp. 138–147.

[6] A. E. Stevens *et al.*, "A time-to-voltage converter and analog memory for colliding beam detectors," *IEEE J. Solid-State Circuits*, vol. 24, no. 6, pp. 1748–1752, Dec. 1989.

[7] S. Tragoudas and N. Denny, "Testing for path delay faults using test points," in *Proc. Int. Symp. Defect and Fault Tolerance VLSI Syst.*, Nov. 1999, pp. 86–94.

[8] S. Wang, X. Liu, and S. T. Chakradhar, "Hybrid delay scan: A low hardware overhead scan based delay test technique for high fault coverage and compact test sets," in *Proc. Des., Autom. and Test Eur.*, 2004, pp. 1296–1301.

[9] M. Nakao *et al.*, "Low overhead test point insertion for scan-based BIST," in *Proc. Int. Test Conf.*, 1999, pp. 348–357.

[10] R. Datta, A. Sebastine, A. Raghunathan, and J. A. Abraham, "On-chip delay measurement for silicon debug," in *Proc. Great Lakes Symp. VLSI Syst.*, 2004, pp. 145–148.

[11] C. Su, Y.-T. Chen, M.-J. Huang, G.-N. Chen, and C.-L. Lee, "All digital built-in delay and crosstalk measurement for on-chip buses," in *Proc. Des., Autom. and Test Eur.*, 2000, pp. 527–531.

[12] K. Heragu, J. H. Patel, and V. D. Agrawal, "Segment delay faults: A new delay fault model," in *Proc. VLSI Test Symp.*, 1996, pp. 32–39.

[13] J. M. Rabaey, *Digital Integrated Circuits*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[14] B. Razavi, *Design of Analog CMOS Integrate Circuits*. Boston, MA: McGraw Hill.

[15] H. Banba, H. Shiga, A. Umezawa, T. Miyaba, T. Tanzawa, S. Atsumi, and K. Sakui, "CMOS bandgap reference circuit with sub-1-V operation," *IEEE J. Solid-State Circuits*, vol. 34, no. 5, pp. 670–674, May 1999.

[16] Sarpeshkar *et al.*, "Mismatch sensitivity of a simultaneously latched CMOS sense amplifier," *IEEE J. Solid-State Circuits*, vol. 26, no. 10, pp. 1413–1422, Oct. 1991.

[17] K. Heragu, J. H. Patel, and V. D. Agrawal, "SIGMA: A simulator for segment delay faults," in *Proc. Int. Conf. Comput.-Aided Des.*, 1996, pp. 502–508.

[18] F. Brglez, "On testability of combinational networks," in *Proc. Int. Symp. Circuits Syst.*, 1984, pp. 221–225.

[19] Synopsys Inc., *Tetramax ATPG Userguide*. [Online]. Available: www.synopsys.com/products

Swaroop Ghosh received the B.E. degree in electrical engineering from the University of Roorkee (now IIT, Roorkee), Roorkee, India, in 2000 and the M.S. degree from the University of Cincinnati, Cincinnati, OH, in 2004. He is currently working toward the Ph.D. degree at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

From 2000 to 2002, he was with Mindtree Technologies Pvt. Ltd., Bangalore, India as a VLSI Design Engineer. His research interests include digital testing and low-power and robustcircuit design for nanometer technologies.

Swarup Bhunia (S'00–M'05) received the B.E. (with honors) from Jadavpur University, Kolkata, India, the M.Tech. degree from Indian Institute of Technology, Kharagpur, India, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 2005.

He is currently an Assistant Professor of electrical engineering and computer science with Case Western Reserve University, Cleaveland, OH. He has worked in the semiconductor industry on RTL synthesis, verification, and low-power design for about three years.

Dr. Bhunia was a recipient of the SRC Technical Excellence Award in 2005, the Best Paper Award at the International Conference on Computer Design in 2004, the Best Paper Award at the Latin American Test Workshop in 2003, and a Best Paper nomination at the Asia and South Pacific Design Automation Conference in 2006. He has served in the technical program committee of Design Automation and Test in Europe 2006 and on the program committee of the International Online Test Symposium 2005. He is currently serving as a program committee member of the Test Technology Educational Program.

Arijit Raychowdhury (M'02) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Calcutta, India, in 2001. He is currently working toward the Ph.D. degree in electrical and computer engineering at Purdue University, West Lafayette, IN.

He was an Analog Circuit Designer with Texas Instruments Incorporated, India. His research interests include device/circuit design for scaled silicon and nonsilicon devices.

Mr. Raychowdhury was a recipient of the Academic Excellence Award in 1997, 2000, and 2001, and the Messner Fellowship from Purdue University in 2002, and the Best Student Paper Award at the IEEE Nanotechnology Conference in 2003.

Kaushik Roy (S'83–M'83–SM'95–F'02) received the B.Tech. degree in electronics and electrical communications engineering from the Indian Institute of Technology, Kharagpur, India, and the Ph.D. degree from the University of Illinois, Urbana-Champaign, in 1990.

He was with the Semiconductor Process and Design Center, Texas Instruments Incorporated, Dallas, TX, where he worked on field-programmable field array architecture development and low-power circuit design. He joined the electrical and computer engineering faculty, Purdue University, West Lafayette, IN, in 1993, where he is currently a Professor, the Roscoe H. George Professor of Electrical and Computer Engineering and a Purdue University Faculty Scholar. He is also the Chief Technical Advisor of Zenasis Inc., Campbell, CA. He was a Research Visionary Board Member with Motorola Laboratories in 2002. He has published more than 350 papers in refereed journals and conferences. He is a coauthor of two books on *Low Power CMOS VLSI Design* (Hoboken, NJ: Wiley, 2000; New York: McGraw-Hill, 2004) and holds eight patents. His research interests include very large scale integration (VLSI) design/computer-aided design for nanoscale silicon and nonsilicon technologies, low-power electronics for portable computing and wireless communications, VLSI testing and verification, and reconfigurable computing.

Dr. Roy was a recipient of the National Science Foundation Career Development Award in 1995, the IBM Faculty Partnership Award, the ATT/Lucent Foundation Award, the 2005 SRC Technical Excellence Award, the SRC Inventors Award, and the Best Paper Awards at the 1997 International Test Conference, the IEEE 2000 International Symposium on Quality of IC Design, the 2003 IEEE Latin American Test Workshop, the 2003 IEEE Nano, the 2004 IEEE International Conference on Computer Design, and the 2005 IEEE Circuits and System Society Outstanding Young Author Award (Chris Kim). He has been on the Editorial Board of the *IEEE Design and Test of Computers*, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and the IEEE TRANSACTIONS ON VLSI SYSTEMS. He was a Guest Editor for the Special Issue on Low-Power VLSI in the *IEEE Design and Test of Computers* (1994), the IEEE TRANSACTIONS ON VLSI SYSTEMS (June 2000), and the *IEE Proceedings—Computers and Digital Techniques* (July 2002).