Local At-Speed Scan Enable Generation for Transition Fault Testing Using Low-Cost Testers

Nisar Ahmed, *Student Member, IEEE*, Mohammad Tehranipoor, *Member, IEEE*, C. P. Ravikumar, *Senior Member, IEEE*, and Kenneth M. Butler, *Senior Member, IEEE*

***Abstract*—At-speed testing is becoming crucial for modern very-large-scale-integration systems, which operate at clock speeds of hundreds of megahertz. In a scan-based test methodology, it is common to use a transition delay fault model for at-speed testing. The launching of the transition can be done either in the last cycle of scan shift [launch-off-shift (LOS)], or in a functional launch cycle that follows the scan shift and precedes the fast capture [launch-off-capture (LOC)]. The LOS technique offers significant advantages over the LOC in terms of coverage and pattern count, but since it requires the scan enable (SEN) signal to change state in the time period of one functional clock cycle, considerable engineering resources are required to close the timing on the SEN signal. Low-cost testers will not be able to provide the at-speed SEN signal as required by the LOS technique. We propose a scan-based at-speed methodology that generates "local" SEN signals that are guaranteed to switch in one functional clock cycle even when the external SEN signal does not change state at functional speed. Our technique is based on encapsulating the SEN control signal in the scan test data. A new scan cell, which is called the last transition generator, must be inserted in every scan chain for generating internal SEN signals. The proposed method is robust, practical, and readily implemented using commercial tools available today.**

***Index Terms*—Launch-off-capture (LOC), launch-off-shift (LOS), low-cost tester, scan enable signal (SEN), transition delay fault testing.**

I. Introduction

THE semiconductor industry is adopting new fabrication processes to reduce the cost per chip and meet the area, power, and performance requirements. As a result, modern ICs are growing more complex in terms of gate count and operating frequency [1]. The defects seen in higher technology nodes ($\geq$ 130 nm) were mostly static in nature, which can be targeted and detected by traditional stuck-at tests. In addition to stuck-at tests, I_{DDQ} testing became an effective way to increase test quality. This was because a quiescent leakage current in CMOS was low enough for an unusually high leakage to be able to detect many of the nonstatic defects.

As technology scales, test engineers are facing new challenges. Over the years, test methodologies for digital circuits have evolved along with technology. The deep-submicrometer (DSM) effects are becoming more prominent with shrinking technology, thereby increasing the probability of timing-related defects [2], [3]. For DSM designs, the stuck-at and I_{DDQ} tests alone cannot ensure a high quality level of chips because of the increasingly growing number of timing-related defects in a chip. In the past, functional patterns were used for at-speed test. However, functional testing is not a viable solution because of the difficulty and time to generate these tests for complex designs with very high gate density. Therefore, more robust at-speed techniques are necessary as the number of timing-related defects is growing and effectiveness of functional and I_{DDQ} testing is reducing [4], [5].

Scan-based structural tests generated by an automatic test pattern generator (ATPG) are increasingly used as a cost-effective alternative to the at-speed functional pattern approach by providing high controllability and observability [5]. The transition fault and path delay fault testing together provide relatively good coverage for delay-induced defects [6], [7]. The path delay model targets the cumulative delay through the entire list of gates in a predefined path while the transition fault model targets each gate output in the design for a slow-to-rise and slow-to-fall delay fault [8]. The transition fault model is more widely used than path delay because it tests for at-speed failures at all nets in the design, and the total fault list is equal to twice the number of nets. On the other hand, there are billions of paths in a modern design to be tested for path delay fault leading to analysis effort; this makes the path delay fault model very cost intensive compared to transition fault model.

A transition fault test requires a pattern pair $(V1, V2)$ to be applied to the circuit-under-test (CUT). Pattern $V1$ is termed as the initialization pattern and $V2$ as the launch pattern. The response of the CUT to the pattern $V2$ is captured at the operational functional speed. The entire operation can be divided into three cycles: 1) Initialization cycle (IC), where the CUT is initialized to a particular state ($V1$ is applied); 2) launch cycle (LC), where a transition is launched at the target gate terminal ($V2$ is applied); and 3) capture cycle (CC), where the transition is propagated and captured at an observation point.

Depending on how the transition is launched and captured, there are three transition fault pattern generation methods. In the first method, referred to as launch-off-shift (LOS) or skewed load [9], the transition at the gate output is launched in the last

Manuscript received July 9, 2005; revised November 17, 2005 and February 17, 2006. The work of M. Tehranipoor was supported in part by the SRC under Grant 2006-TJ-1455. This paper was recommended by Associate Editor K. Chakrabarty.

N. Ahmed is with the University of Connecticut, Storrs, CT 06269 USA (e-mail: nisar@engr.uconn.edu).

M. Tehranipoor is with the Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269 USA (e-mail: tehrani@engr.uconn.edu).

C. P. Ravikumar is with the Texas Instruments, India (e-mail: ravikumar@ti.com).

K. M. Butler is with the Texas Instruments, Dallas, TX 75243 USA (e-mail: kenb@ti.com).

Digital Object Identifier 10.1109/TCAD.2006.884405

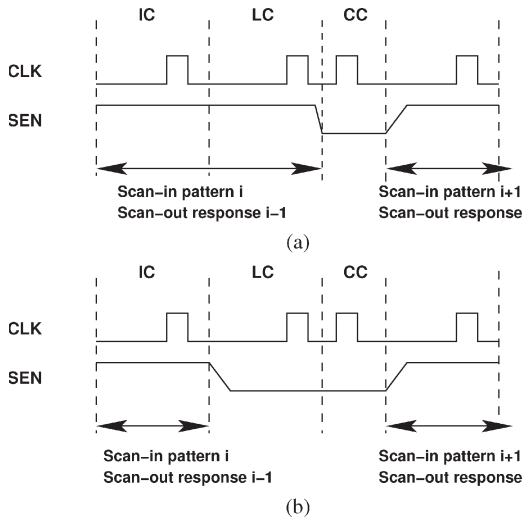

Fig. 1. Transition delay fault pattern generation methods: (a) LOS and (b) LOC.

shift cycle during the shift operation. Fig. 1(a) shows the LOS method waveform for a multiplexed-D flip–flop (DFF) design; a similar approach can be applied to an level sensitive scan design (LSSD). The LC is a part of the shift operation and is immediately followed by a fast capture pulse. The scan enable (SEN) is high during the last shift and must go low to enable a response capture at the CC clock edge. The time period for SEN to make this $1 \rightarrow 0$ transition corresponds to the functional frequency. Hence, the LOS requires the SEN signal to be timing critical. Skewing the clock (CLK) creates a higher launch-to-capture clock frequency than standard shift clock frequency. Saxena *et al.* [12] list more launch and capture waveforms used by the LOS approaches. In [12], implementation of scan-based transition fault testing and its low-cost test challenges are discussed.

Fig. 1(b) shows the waveforms of the second approach, referred to as launch-off-capture (LOC) or broadside method [10]. In this method, the LC is separated from the shift operation. At the end of scan-in (shift mode), pattern $V1$ is applied and the CUT is set to an initialized state. A pair of at-speed clock pulses is applied to launch and capture the transition at the target gate terminal. This relaxes the at-speed constraint on the SEN signal, and dead cycles (DCs) are added after the last shift to provide enough time for the SEN signal to settle low.

The third technique, known as "enhanced scan" [11], requires that two vectors $V1$ and $V2$ are shifted into the scan flip–flops simultaneously. The drawback on enhanced scan is that it needs hold-scan flip–flops and is area intensive [11], making it unattractive for application specific integrated circuit (ASIC) designs.

The LOS method is more preferable based on the ATPG complexity and pattern count compared to LOC method. The LOC technique is based on a sequential ATPG algorithm, while the LOS method uses a combinational ATPG algorithm. This will increase the test pattern generation time in case of LOC, and also, a high fault coverage cannot be guaranteed due to the correlation between the two patterns, $V1$ and $V2$; note that $V2$ is the functional response of pattern $V1$. Due to its dependence on the functional capture path for launching transitions, LOC ATPG can be quite slow and on large designs can consume several days of ATPG runtime. The main concern about the LOS is its requirement to at-speed SEN signal.

A. Prior Work

As design sizes grow due to higher integration, the number of flip–flops in a design is increasing, and it is not uncommon to see designs with several hundred thousand flip–flops today. The physical design of the SEN signal for LOS is therefore as challenging as that of the clock signal. An approach similar to clock tree synthesis is not used to address this issue due to the associated design cost [13]. The problem can be alleviated by increasing the number of SEN ports, thereby reducing the fan-out of SEN signal, but this is not practical when low-cost testers are employed [12].

In [14], a hybrid architecture is proposed which controls a small subset of selected scan cells by LOS, and the rest are controlled by the LOC approach. A fast SEN signal generator is designed which drives the LOS-controlled scan flip–flops. Its design is such that the fast SEN signal makes a transition only on the negative edge of the controlling clock. Therefore, all positive-edge scan flip–flops lose half a cycle for the SEN signal. Moreover, the selection criteria of the LOS-controlled scan flip–flops and the order in which these flip–flops are stitched determine the effectiveness of the method. In some cases, the number of patterns generated by the hybrid method exceeds the LOC pattern count [14].

A widely used method in industrial practice is to pipeline the SEN signal [15]. In a multistage pipeline SEN, the designer must carefully select the group of scan cells controlled by the respective SEN signal. In order to meet timing closure of the pipeline SEN signals, these selection criteria manifest into the design iterations and additional design effort for the place and route (PNR) tool especially for scan reordering during routing step.

Several other researchers have also focused on other issues related to transition fault testing, such as fault coverage, ATPG, yield loss, functionally testable and untestable faults, etc. Bhunia *et al.* [16] proposed an enhanced scan-based delay fault testing which reduces the area overhead when compared to conventional enhanced scan [11]. However, the proposed technique offers high area and delay overhead compared to LOC and LOS methods. Instead of using an extra latch, a supply gating was proposed to be used at the first level of logic gates of a combinational circuit.

A transition fault ATPG methodology flow for scan-based designs using broadside (LOC) test format is proposed in [17]. A replicate and reduced circuit transform maps the two-time frame processing of the transition fault ATPG to a single-time frame processing on duplicated iterative blocks with reduced connections. Abadir and Zhu [17] report high transition fault coverage using this ATPG method.

A path-oriented test generation procedure called POTENT is proposed in [18] to generate high-quality tests for transition faults. Conventional ATPG tools are unaware of the circuit states exercised during the functional operation of the circuit.

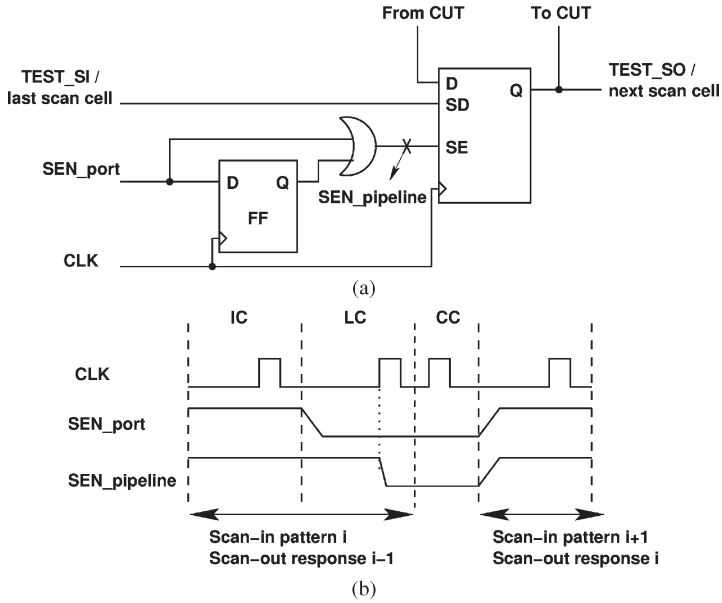

Fig. 2. Pipeline SEN.

In [19], Liu and Hsiao presented a concept of testing only functionally testable transition faults in broadside transition testing via a constrained ATPG. The method generates a test set for functionally testable transition faults and minimizes detection of functionally untestable transition faults by the assumption that the ATPG knows illegal states.

A SAT-based ATPG for path-oriented transition faults is proposed in [20]. A transition fault is detected through the longest path. A false path pruning technique is utilized to identify the longest path through each fault site, which results in a faster test pattern generation. Rearick [21] argues that the scan-based transition fault testing is shown to allow tests to be applied which are unrealizable in normal operation. In other words, scan-based fault testing methods have a negative impact on yield and designer productivity due to overtesting for transition faults.

B. Contribution and Paper Organization

In this paper, we propose a scheme to practice the LOS methodology using a low cost/speed tester. The SEN control information for the launch and CC is embedded in the test data itself. A new scan cell, called the last transition generator (LTG), generates the local fast SEN signals used to control scan cells in a scan chain. The LTG cell has the flexibility to be inserted anywhere in the scan chain, and the hardware area overhead is comparable to the pipeline SEN approach. Unlike the previous arbitrary selection of scan flip–flops across different scan chains controlled by the local SEN (LSEN) signal, the LTG cells drive a set of scan flip–flops in a scan chain. This will result in a much lower signal enable routing area. The proposed method poses no additional constraints for the PNR tool, and the scan flip–flop selection method provides a better mechanism to reorder the scan cells to meet the timing closure of the LSEN signals.

The rest of this paper is organized as follows. The pipeline SEN method, its constraints during the scan insertion, and the motivation are discussed in Section II. Section III describes the LSEN generation and the architecture and operation of the LTG cell. Section IV explains the test architecture and LOS clock timing waveforms. The discrete Fourier transform (DFT) insertion and test protocol file changes for ATPG along with the experimental results are discussed in Section V. Finally, concluding remarks are given in Section VI.

II. Background and Motivation

The main concern in implementing the LOS method is that it requires the SEN signal to switch at functional speed. Here, we describe the pipelined SEN methodology that is able to overcome this problem. Fig. 2 shows the basic implementation and operation of a single stage pipeline SEN for LOS [15]. The SEN port (SEN_port) is deasserted asynchronously in the IC. The pipeline SEN (SEN_pipeline) goes low synchronously at the active edge of launch clock cycle. The SEN_port is asserted high after the CC to re-enter the shift mode. The SEN_pipeline signal is generated synchronously and has one full clock cycle available for transition, assuming the pipeline SEN signal is generated separately for positive- and negative-edge flip–flops. Also, the tester skew at the SEN port is decoupled from the internal SEN net.

Fig. 3 shows a three-stage pipeline SEN architecture. A leaf pipeline stage is defined as the last stage of the SEN pipeline tree which directly drives the SEN of a subset of scan flip–flops in the respective logic cone. There are mainly two concerns in the pipeline SEN insertion: 1) determine the scan flip–flops to be controlled by each leaf stage and 2) determine the level (module or top) at which the leaf stages are inserted. Earlier, the test engineers used to arbitrarily select the controlled scan flip–flops. In arbitrary selection, the leaf-stage pipeline will control the scan flip–flops across different scan chains. The physical design tool is challenged in two dimensions: 1) re-order scan flip–flops in respective scan chains to reduce each scan-chain routing area and 2) placement of the leaf pipeline

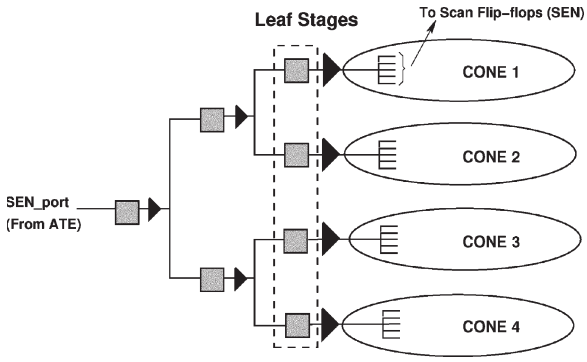

Fig. 3. Pipeline SEN architecture.

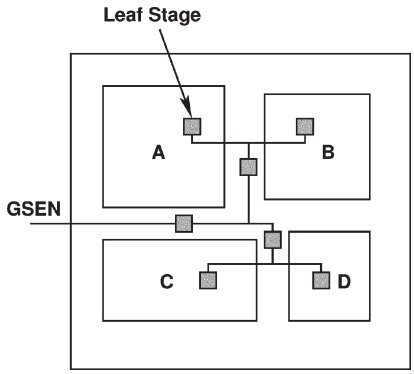

Fig. 4. One possible pipeline SEN insertion technique.

SEN stage to reduce the area (repeater + routing) across chains (arbitrary selection might select flip–flops across scan chains). Selection of flip–flops in the same scan chain will reduce this twofold problem into one, and the reorder of scan cells can be used in a positive sense to reduce the SEN routing also.

Typically, in very large designs, scan insertion is performed at module level and, later, the module level scan chains are stitched at the top level. Customarily, one pipeline stage is added per module and, later, at the top level, these are connected in the form of a tree network. Fig. 4 shows this SEN pipeline insertion methodology. In this example, there are four blocks A, B, C, and D, and each block is inserted with a pipeline stage which forms the leaf stage of the SEN tree. The leaf stages are connected with additional stages at the top level depending on the timing requirements. Since each module contains different number of flip–flops, the number of scan cells driven by the leaf stages will be different, resulting in an imbalanced SEN tree. Also, different clock domains have different clock insertion delays; the unbalanced tree may result in additional effort for physical designers to adjust the drive strengths of buffers in different branches of the tree.

In modern-day very-large-scale-integration (VLSI) design flows, the physical design flow reorders the scan flip–flops in scan chains to reduce the routing congestion, after placement step based on the proximity of the scan flip–flops. Since the leaf stages control the scan flip–flops across different scan chains, the reordering of scan flip–flops may negatively impact the routing of the SEN signal. The problem however is to determine the set of scan flip–flops controlled by each pipelined SEN stage so that proximity of flip–flops to the leaf stage is taken into account.

In this paper, we propose a generation of fast SEN signals through the use of some special sequential cells called LTG cells; these LTG cells are similar to scan flip–flops, and one or more LTG cells are inserted into a scan chain. The LTG cell generates the SEN signals for all or a subset of cells in the same scan chain. A physically aware scan-insertion tool can treat the LTG cells similar to scan flip–flops and can hence ensure proximity of the LTG cell to all the flip–flops that it controls. This is straightforward when we have a single LTG cell per scan chain. Sometimes, when the scan chain has a large number of flip–flops, more than one LTG cell may become necessary. In this case, we can still use the scan-insertion tool to first create smaller scan segments, each of which contains one LTG cell, and then stitch the scan segments together. In modern system-on-a-chip (SoC) designs that employ test data compression, it is common to use a large number of short scan chains [24]. The proposed method will be a natural fit in such situations. Moreover, in our technique, the leaf stages are inserted in the scan path which increases the testability of the design and inserted cells. Note that the leaf stages in a pipelined SEN cannot be easily tested unlike our proposed technique's cells inserted in the scan chain.

III. LSEN Signal Generation

In order to improve testability and provide more flexibility with all the advantages of the pipeline SEN, the LSEN generation method is proposed such that SEN generator cells are inserted within the scan chains. Therefore, the control information can be passed as part of the test data. Before we describe our proposed LSEN generator cell, we first explain how the transition data can be passed through the scan chain as part of the test data without inserting an LTG cell in the scan chain. Fig. 5 shows a small example of generating the LSEN signal from the test pattern data for LOS. The external SEN signal from the tester is referred to as the global SEN (GSEN). The internally generated SEN signal is termed as LSEN. The main objective is to deassert the GSEN in the IC without the at-speed constraint, and then generate the at-speed LSEN signal during the launch and CC synchronously from the test data. There are eight scan flip–flops in the scan chain, and the test pattern shifted is $b_8b_7b_6b_5b_4b_3b_2b_1 = 10001110$, assuming b_1 is the first bit shifted into the scan chain. The values shifted into the scan flip–flops during the various shift cycles are shown under each flip–flop. GSEN is deasserted during the $(n-1)$th shift cycle (IC), where $n = 8$.

For proper shift operation, the LSEN signal should be logic 1 in the $(n-1)$th cycle of the shift operation (IC) and logic 0 in the last shift cycle (LC) to enable capture in the next clock cycle. In other words, the LSEN signal must make a $1 \rightarrow 0$ transition at the launch edge. For this particular example, the pattern during the shift operation generates the required $1 \rightarrow 0$ transition at the output of scan flip–flop A. The output of scan flip–flop A is ORed with GSEN to generate the LSEN signal. Therefore, the final value of scan flip–flop (A) and its following scan flip–flop (b_5 and b_4) at the end of shift operation

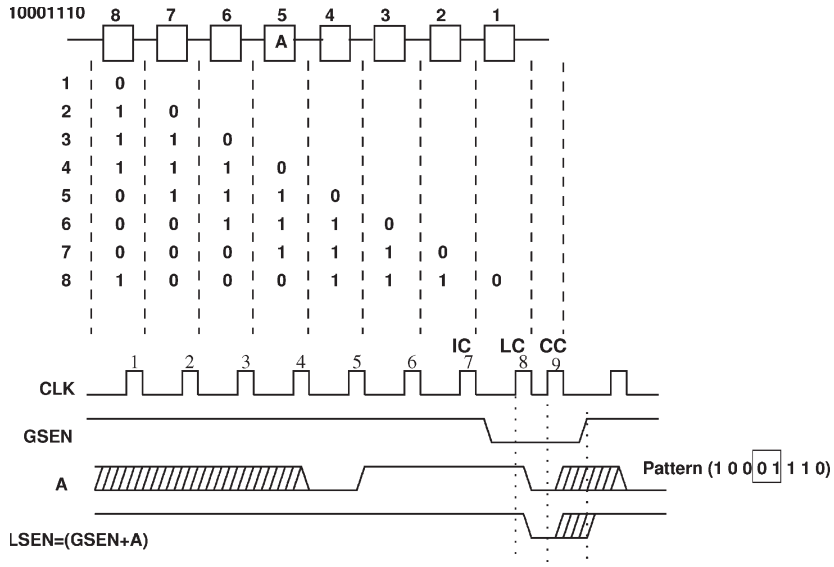

Fig. 5. Generation of LSEN signal.

must be 0 and 1, respectively, so that A is loaded with logic 1 in IC and logic 0 in LC. A full functional clock cycle is available for LSEN to make the transition. After the CC, the LSEN signal is asynchronously set to 1 by GSEN for scanning out the response.

A. LTG

As explained earlier, during the LOS pattern generation, to generate the SEN transition $1 \rightarrow 0$ at the launch edge, the scan flip–flop A should be 0 and the following scan flip–flop must be 1. This is very unlikely to occur in every pattern generated during ATPG. It can also be seen in Fig. 5 that there is an unknown value loaded in A during the capture edge, which can cause a problem if the method is to be used for clock sequential pattern generation with multiple CCs. For such patterns, the LSEN signal must be deasserted and multiple capture clocks must be applied. Applying a capture signal may also change the content of the scan flip–flop A since it is connected to the CUT. As a result, LSEN may change depending upon what value (0 or 1) is received as a response bit into the scan flip–flop A. Hence, A must be constrained to 0 during ATPG during pattern generation, and the scan flip–flop A must be selected such that the response coming back into A be 0. This constraint is necessary but not sufficient for proper operation as the value loaded in A from the functional path after applying the first system clock is not known. Therefore, after going to capture control state (0), the LSEN must remain in this state as long as it is asynchronously set to 1 by GSEN. This requires additional ATPG constraints for a conventional scan flip–flop architecture to control the functional path (D) to logic 0. This might lead to a fault coverage reduction.

In order to avoid loss of coverage without a significant change in the architecture, a new scan cell called LTG is designed such that when the scan cell is loaded with logic 0 during pattern shift and the SEN control is made logic 0, i.e., capture state, the cell will remain in that state until it is shift controlled by the GSEN. This cell is inserted into the scan chain

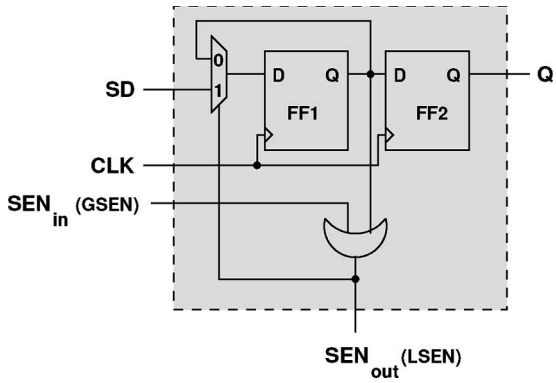

Fig. 6. LTG cell.

to generate the $1 \rightarrow 0$ transition at the launch edge through the scan path.

Fig. 6 shows the LTG cell architecture. It consists of two flip–flops which are used to load the control information required for the launch and CCs. The port definition is similar to a scan cell, and the output of FF1 (Q) is fed back to the functional input port of the scan cell. The input D of LTG cell does not exist because it does not connect to CUT. The LTG cell consists of an input SEN $(\mathrm{SEN_{in}})$ pin which takes the GSEN signal as input. An additional output SEN $(\mathrm{SEN_{out}})$ pin (GSEN + Q) represents the LSEN signal. FF2 does not allow the output of FF1 to be shifted into the next normal scan cell in scan chain in the last shift process. The LTG cell can be inserted anywhere in the scan chain, and it is not connected to the CUT. Therefore, any ATPG constraint on the LTG cell does not affect the CUT fault coverage.

Theorem: The LSEN signal generated by the LTG cell switches at-speed during the CC.

Proof: $\mathrm{SEN_{out}}$ refers to the LSEN signal in the LTG cell of Fig. 6. The clock input to the LTG cell for LOS transition delay ATPG is of the form shown in Fig. 1(a). It is assumed that the clock tree synthesis tool is capable of routing the clock signal so that the local clock signal at the input of the LTG cell switches at functional speed during the LC and CC cycles.

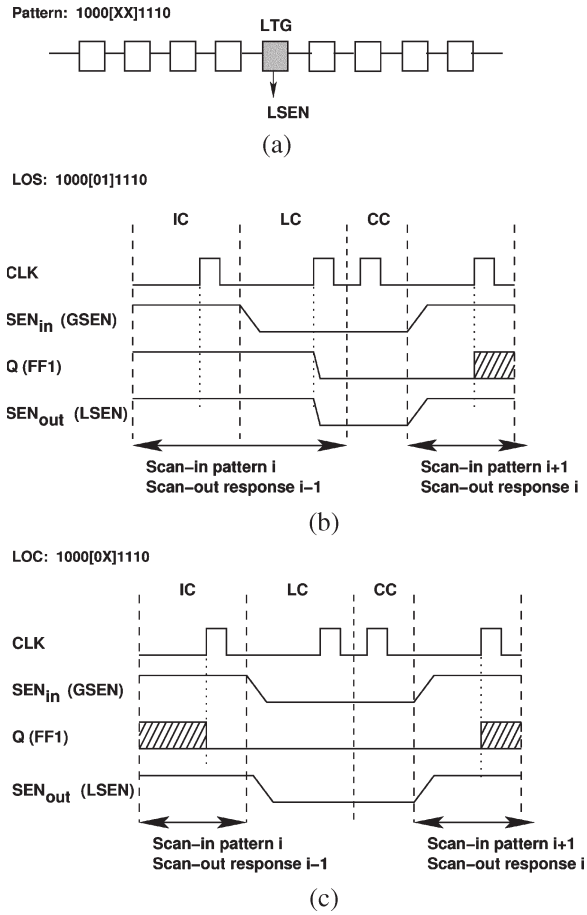

Fig. 7. Operation of LTG cell. (a) Example scan chain. (b) LOS. (c) LOC.

During the scan shift cycle (IC), a "1" is scanned into the flip–flop FF1 of LTG cell at low frequency. During the last cycle of shift, denoted by LC, the clock switches at functional speed and the output of FF1 also switches to 0 state at the functional speed, since the number of flip–flops driven by $\mathrm{SEN_{out}}$ is an order of magnitude smaller than the total number of flip–flops in the design, thereby reducing the capacitive load on the LSEN signal. The GSEN signal switches to 0 during the beginning of the LC cycle. Let A refer to the output of FF1. $\mathrm{SEN_{out}}$ is the logical OR of the signal A and the GSEN signal; therefore, $\mathrm{SEN_{out}}$ also switches at the speed of signal A, except for the small delay in the OR gate. ■

B. Operation of LTG Cell

Fig. 7(a) shows the previous example with the LTG cell inserted in the scan chain. Note that the LTG cell can be placed anywhere in the scan chain, and it is not connected to the CUT. The values of the two flip–flops of the LTG cell in the test pattern are shown as X (1000[XX]1110). These flip–flops are constrained during ATPG to assign specific values to the Xs. Fig. 7(b) shows the pattern and the timing waveform for LOS. During the shift operation, at the last shift, the SEN must make a $1 \rightarrow 0$ transition. Thus, FF1 of LTG cell should be loaded with 1 in IC, followed by 0 in the next cycle (LC). The $\mathrm{SEN_{in}}$ (GSEN) signal is asynchronously deasserted in IC. The $\mathrm{SEN_{out}}$ (LSEN) signal is generated by the Boolean equation $\mathrm{SEN_{out}} = A(Q_{\mathrm{FF1}}) + \mathrm{SEN_{in}}$. After the CC, the LSEN is asserted to 1 asynchronously by GSEN.

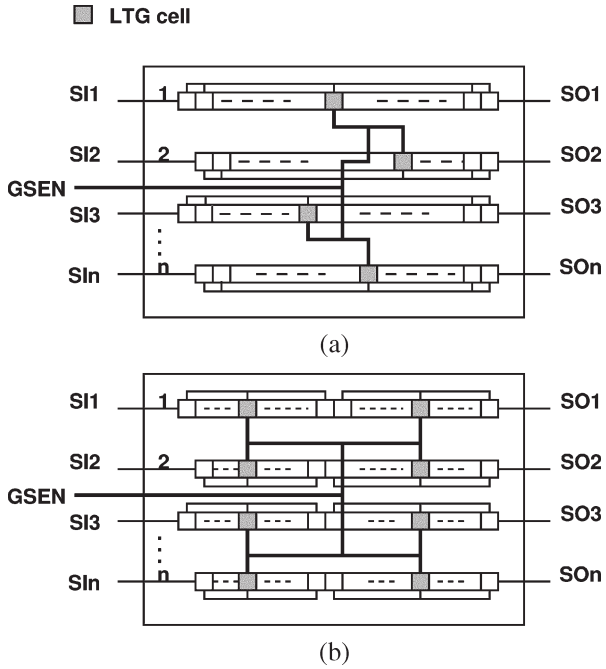

Fig. 8. Test architecture. (a) One LTG cell per scan chain. (b) Two LTG cells per scan chain.

For LOC, the GSEN signal is high during the entire shift operation, after which, the GSEN is asynchronously deasserted and LSEN must change to logic 0. For LOC, since the LSEN must remain 0, the value in FF2 is a don't-care. The value of FF1 of LTG cell must be constrained to 0 during ATPG. Fig. 7(c) shows the pattern and the timing waveform for LOC. It can be noticed that $\mathrm{SEN_{out}}$ switches at the speed of GSEN (not at-speed).

Compare the LOS timing waveforms in Figs. 1(a) and 7(b). In Fig. 1(a), the SEN signal may feed both positive-edge and negative-edge triggered flip–flops. In the worst case, therefore, the SEN signal must make a transition at the negative edge of the LC clock. This cuts down the time for SEN transition to a half of the functional clock cycle, making the timing closure harder. In the timing waveform of Fig. 7(b), we see that this problem can be eliminated. It is customary to place positive-edge and negative-edge flip–flops on separate scan chains. Therefore, a separate LTG cell can be used to control posedge and negedge flip–flops. As a result, every LSEN signal has one complete functional clock cycle to transition.

IV. DFT Architecture

In conventional scan architecture, the GSEN is connected to all scan cells in every scan chain. While in the new architecture, the local at-speed SEN (LSEN), generated by LTG cell using the GSEN and pattern data, is connected to each and every scan cell in the respective scan chain. The GSEN signal is only connected to LTG cells. In general, there can be multiple scan chains in a design to reduce the test application time. Fig. 8(a) shows a multiple scan-chain architecture with n scan chains, and each scan chain contains m scan cells. As shown, each

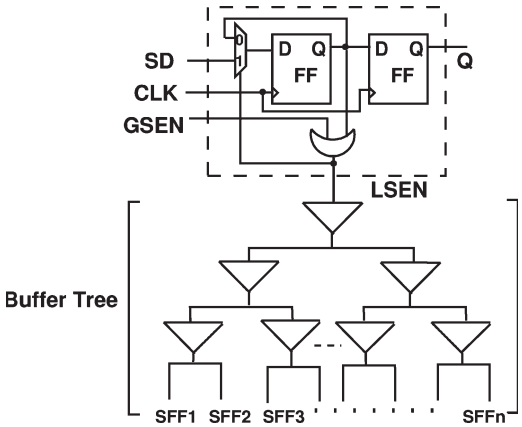

Fig. 9. LTG cell with buffer tree driving SEN signal.

scan chain i, where $1 \leq i \leq n$, consists of one LTG cell which generates the fast SEN signal LSEN_i for m scan cells.

The area overhead of an LTG cell (equivalent to the sum of two flip–flops and some glue logic) is a few extra gates, which is negligible in modern designs. Note that if the SEN timing is not met, then multiple LTG cells can be inserted to generate multiple LSEN signals to control different segments of the same scan chain as shown in Fig. 8(b). In this case, we design two smaller trees for a local enable signal.

The main area overhead of a tree design is the routing area and buffers inserted on the tree. The fan-out load on the GSEN signal is reduced, and the fan-out load driven by the LSEN signal is used as a constraint to find the number of LTG cells inserted. For example, for k total number of flip–flops in a design and l being the maximum number of flip–flops that can be allowed for the LSEN to be timing closed for a particular operating frequency, the number of LTG cells are estimated by k/l.

Although the area of the LTG cell alone is negligible, the size of the buffer tree driving the LSEN varies depending on the number of controlled scan flip–flops. Note that, in general, the buffer tree is required for all cases when using: 1) GSEN as an at-speed SEN signal generated using automatic test equipment (ATE); 2) the pipeline method; and 3) the LTG-based method. For example, when using an at-speed GSEN derived from tester, a clock tree is required to connect the GSEN to all the scan flip–flops in the design. The size of the tree depends on the drive strength of the signal and the number of scan flip–flops.

Fig. 9 shows the architecture of an LTG cell and the buffer tree driving the LSEN signal. The buffer size and the number of stages required are mutually exclusive. In our experiment, we used a minimum size buffer with a drive strength of two. Fig. 10 shows the number of repeaters in the buffer tree required to drive different number of scan flip–flops. The operating frequency selected was 250 MHz, and a maximum number of 2000 scan flip–flop's SEN, meeting the timing requirements, can be driven by a single LTG cell. As shown in Fig. 10, the number of repeaters required in the buffer tree reduces drastically with a decrease in the number of controlled scan flip–flops.

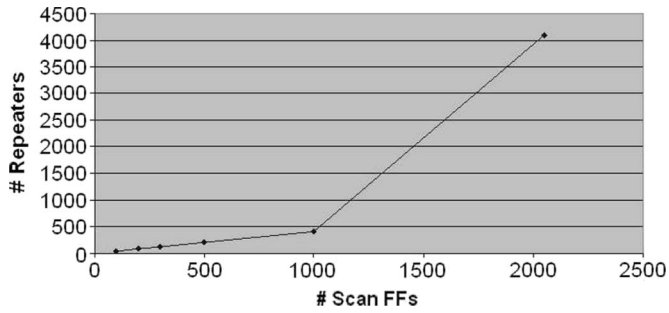

Fig. 10. Number of repeaters required to drive scan flip–flops by a single LTG cell.

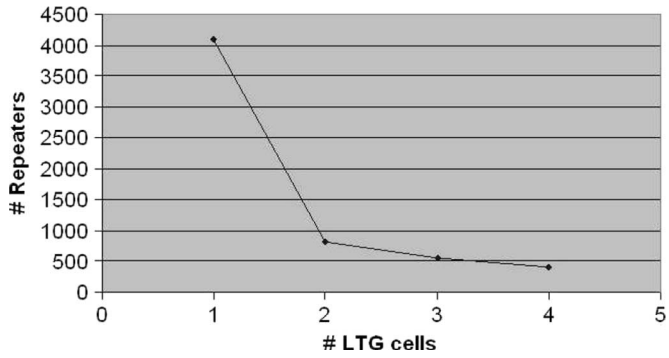

Fig. 11. Number of repeaters required for multiple LTG cells inserted per scan chain (scan-chain $= 2000$).

Fig. 11 shows the relationship between the number of repeaters required and the multiple LTG cell insertion when using 2000 scan flip–flops per scan chain with an operating frequency of 250 MHz. The buffer tree size decreases using multiple LTG cells. The percentage reduction is not significant when inserting two, three, and four LTG cells. Note that further increasing the number of LTG cells may not significantly reduce the buffer tree size while the area overhead of LTG cells will increase. Based on these results, when the scan-chain length $= 2000$ and frequency $= 250$ MHz, it can be recommended to insert at least two to four LTG cells per scan chain. These experiments give only an estimation of the number of LTG cells required and the size of the buffer tree required. Note that, in general, the area of the SEN tree is similar to the pipeline SEN scheme. Also notice that due to increasingly using commercial test compression tools, in practice, the length of scan chains are considerably shorter than 2000. Thus, the size of clock tree to drive the SEN signals will be significantly smaller.

The area overhead can be classified into three different categories: 1) LTG cell; 2) repeaters; and 3) routing. Fig. 12 shows the trend of each of these categories. It can be seen [Fig. 12(a)] that with multiple LTG cell insertion, although the LTG cell area increases (negligible), there is a significant decrease in the repeater area required to drive the LSEN tree. Also, the routing area increases with an increase in the number of repeaters [Fig. 12(b)]. Therefore, an efficient tradeoff would be possible to perform with carefully designing the buffer tree for LSEN and calculating the overall area overhead based on the SEN tree area (number and size of repeaters), routing area, and LTG cells.

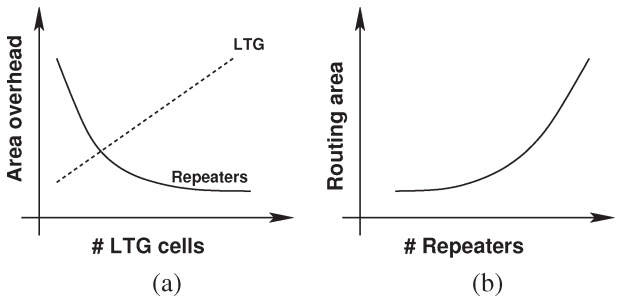

Fig. 12. (a) LTG cell and repeater area overhead. (b) Routing area overhead versus number of repeaters.

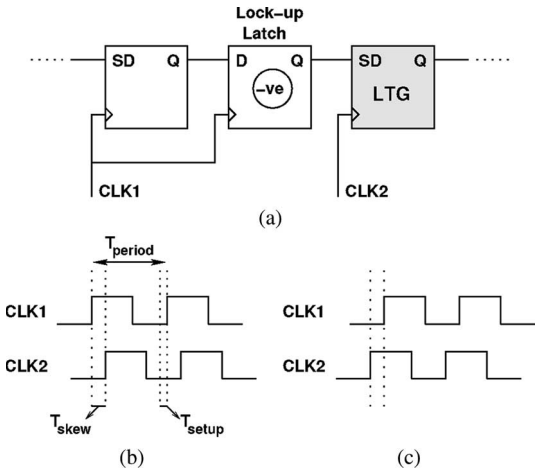

Fig. 13. Multiple clock domain analysis.

A. Multiple Clock Domain Analysis

In general, for a multiclock domain design, the scan chains are clock mixed to balance the scan chains. To avoid setup and hold violations due to interclock domain skews, lockup latches are inserted between flip–flops of different clock domains. Fig. 13 shows an example with two clock domains CLK1 and CLK2, respectively. The LTG cell in the scan chain is connected to CLK2. A negative level sensitive lockup latch is added between the scan flip–flop of CLK1 domain and the LTG cell. This will give an additional half cycle for the other clock domain to avoid skew problems. There are two possible cases: Case1: CLK1 edge occurs before CLK2 [Fig. 13(b)] and Case2: CLK1 edge occurs after CLK2 [Fig. 13(c)]. In Case1, the LTG cell data shifts and the LSEN is generated at CLK2 edge. In Case2, the LSEN is generated at CLK2 edge, which occurs before CLK1. This might result in hold violations in CLK1 domain due to SEN transition. Hence, to avoid hold violations, the LTG cell must be connected to the late arriving clock. The timing constraint for the LSEN signal is that it must be timing closed for the fastest clock domain in the scan chain. It can be represented as

$$T_{\mathrm{SEN}} = T_{\mathrm{period}} - T_{\mathrm{skew}} - T_{\mathrm{setup}}.$$

The transition fault patterns for LOS are generated per clock domain. If the pattern is shifted at a slow speed followed by a fast capture, the time from the launch edge of LC to the capture edge of CC is not really at-speed. Fig. 14(a) shows

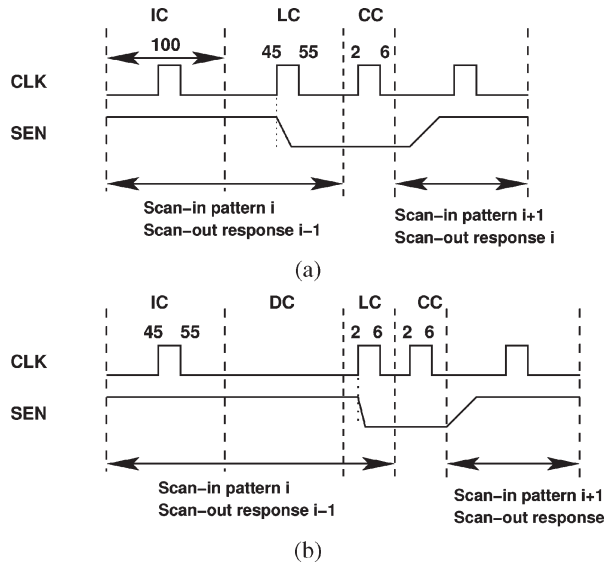

Fig. 14. LOS clock timing waveform.

the limitation of the clock timing waveform. The functional operating frequency is 125 MHz. The launch edge in the last shift occurs at 45 ns, and the capture edge occurs at 2 ns in the CC of 8-ns time period. The time from the launch edge to the capture edge is $(55 + 2) = 57$ ns. Fig. 14(b) shows the modified at-speed clock timing waveform used for LOS. The last shift is done at-speed corresponding to the clock domain being tested, and the capture clock is applied only for that clock domain. A DC is added after the IC for the scan chain to settle down.

B. LTG Insertion Flow

After we determine the number of LTG cells to be inserted based on the optimization analysis explained in the early part of this section, we perform a scan insertion. There are two widely used scan-insertion flows: 1) top-level scan-insertion flow and 2) bottoms-up scan-insertion flow. For smaller designs, the preferred flow is top-level scan insertion, and for large designs, the bottoms-up scan-insertion flow is followed. Fig. 8 shows the top-level scan insertion with a single LTG cell per scan chain.

The DFT insertion tool [23] allows a single internal signal to control the SEN of all the scan flip–flops in a single chain. For a design with N scan chains and M LTG cells to be inserted per scan chain, the scan-insertion tool is directed to insert $N \times M$ scan chains with one LTG cell per scan chain. In the next step, the scan chains are restitched into N scan chains. For example, if $N = 4$ and $M = 2$, there are eight scan chains inserted with one LTG cell per scan chain and, later, these chains are restitched into four chains at top level.

As the designs presently are large and often reuse IPs, the scan insertion is done at module level, and then these chains are connected at the chip level. It is also referred to as bottoms-up approach. Fig. 15 shows the flow 1 where each module is scan inserted, and each scan chain having an individual SEN control signal. The LTG cells are inserted manually between the modules, and the scan chains are stitched at the top level. In flow 2, each modules scan chains are inserted with LTG cells, and the scan chains are stitched at the top level (Fig. 16).

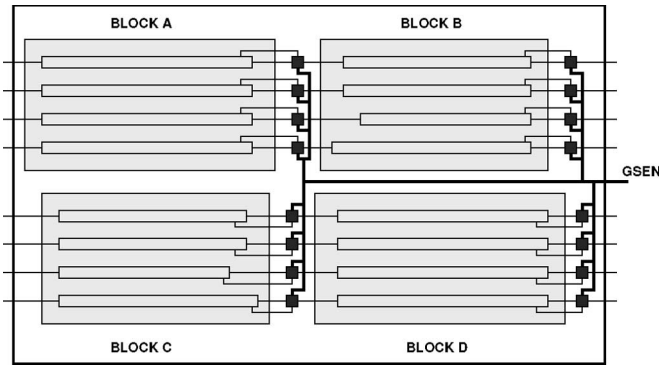

Fig. 15. Bottoms-up scan insertion: Flow 1.

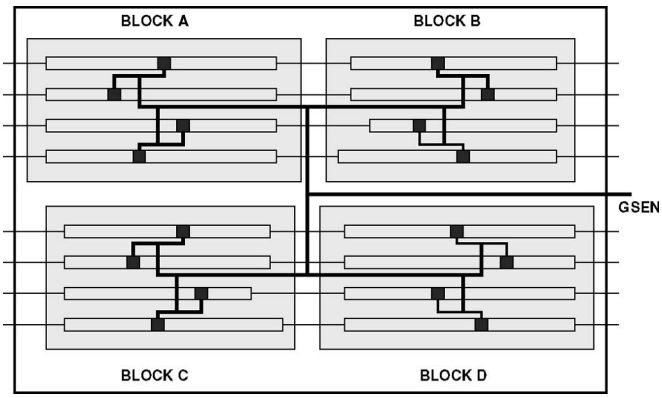

Fig. 16. Bottoms-up scan insertion: Flow 2.

This might be more preferable as the test engineer will only have to hook up the scan chains appropriately and need not worry about the LTG cell insertion.

V. Experimental Results

In this paper, we have argued in favor of the "LOS" transition delay ATPG methodology and presented a technique that can ease the implementation of this technique using low-speed testers. First, we explain how the LTG cells are inserted using a commercial tool and the procedure following during pattern generation. Fig. 17 shows the list of steps required to insert LTG cells during scan-chain insertion. Here, we assume that one LTG cell is inserted per scan chain. The scan-insertion tool needs to recognize the LTG cell as a scan register containing two scan cells in order to stitch it into the scan chain. This requires it to be defined as a new library cell with scan cell attributes. A workaround is to design the LTG cell as a module and declare it as a scan segment (line 04), e.g., set_scan_segment command in Synopsys DFT Compiler [23]. The tool then identifies the LTG cell as a scan segment of length 2. The GSEN signal is connected to all the LTG cell's $\mathrm{SEN_{in}}$ input port, and the clock is connected to CLK input. The scan-insertion tool is then directed to stitch each individual scan chain c_i including LTG_i cell and hook up the $\mathrm{LTG}_i/\mathrm{SEN_{out}}$ port to the SEN port of the remaining scan cells in the chain (line 09). Only the LTG cell needs to be specified in the scan path, as the tool will stitch the rest of the cells including the

DFT:
01: for i = 0 to no_chains
02: {
03: Instantiate LTG_i cell
04: Declare LTG_i cell as scan segment
05: Make connections to LTG cell
06: Connect GSEN
07: Connect clock
08: Stitch scan chain c_i including LTG_i cell
09: Hookup LSEN from LTG_i to all scan cells in the chain
10: }

ATPG:
01: foreach pattern
02: {
03: $N-1$ slow shifts
04: Dead Cycle for GSEN to settle low
05: Nth fast shift
06: Fast capture cycle
07: }

Fig. 17. Scan insertion and ATPG process.

TABLE I
Designs Characteristics

Design	Scan Cells	# Chains	Clocks	LTG cells	TF
A	10477	16	6	16	384998
B	40342	16	4	48	2124502
C	104539	8	1	96	4116092

LTG cell, and balances the scan chain depending on the longest scan-chain length parameter.

The ATPG methodology is no different from conventional when we use the LTG-based solution. The SEN signal for the flip–flops now comes from an internally generated signal. An ATPG tool must be able to place the SEN signal in the active mode during the scan shift operation. Notice that the LTG cell generates the LSEN signal through a logical OR of the GSEN and the Q output of the flip–flop FF1 (see Fig. 6). The $1 \rightarrow 0$ transition of the LSEN is generated by the pattern shift. During "design rule check (drc)" phase, the ATPG tool performs a single sensitization shift where it assumes a don't-care in each individual scan flip–flop and shifts a known value to observe it at the output of each individual scan flip–flop. The tool starts from the scan output and performs this process until it reaches the scan-in pin and checks whether the scan chain is traceable. This makes the ATPG tool to fail in the drc phase as our technique assumes known values "0" and "1" in FF1 and FF2, respectively, to generate the SEN signal internally. In order to get around the problem of the drc phase of the ATPG tool failing due to the internal nature of the SEN signals of the flip–flops, we use a different model of the LTG cell during the ATPG process.

We have experimented on three industrial designs A, B, and C, and Table I shows the characteristics of these designs. In design A, each scan chain is inserted with one LTG cell, while in designs B and C, three and 12 LTG cells are inserted per scan chain, respectively, to meet the SEN timing requirements. The total transition faults are shown in column TF. The test strategy is to get the highest possible test coverage for the transition faults. When generating test patterns for the transition faults, we target only the faults in the same clock domain. During the

TABLE II
RESULTS FOR THREE INDUSTRIAL DESIGNS

Design	FC(%)	# Patt	*Conventional LOS*			*LTG-based LOS*		
			Scan Cells	Max. Length	Test Cycles	Scan Cells	Max. Length	Test Cycles
A	81.44	1093	10477	655	0.715M	10509	657	0.718M
B	84.41	4160	40342	2521	10.487M	40438	2527	10.512M
C	84.16	5958	104539	13067	77.853M	104731	13091	77.996M

pattern generation, only one clock is made active during the CC. Hence, only faults in that particular clock domain are tested. All PIs remain unchanged, and all POs are unobservable while generating the test patterns for the transition faults. This is because the tester is not fast enough to provide the PI values and strobe POs at speed.

The ATPG results and comparison of conventional LOS with LTG-based LOS are shown in Table II. TetraMAX [22] tool was used for ATPG. The FC and # Patt columns show the fault coverage and the number of patterns generated, respectively, for LOS method. The chain length for LTG increases by twice the number of LTG cells inserted per scan chain because each LTG cell contains two flip–flops. The table also shows the area and time overhead of LTG-based LOS method over the conventional LOS method (see columns 7 and 9). Notice that the increase in the number of flip–flops and the test application time by LTG insertion is not very significant.

We believe that the LTG-based solution has the following advantages.

1) The technique can be practiced using existing commercial tools for DFT insertion and ATPG.
2) The technique is applicable to all scan-based DFT techniques, including deterministic built-in self-test (BIST) techniques that are based on scan [24].
3) The method is least intrusive, and it can be incorporated in existing physical design flows.
4) In our experiments, we demonstrated the use of the technique using DFT Compiler and TetraMAX tools from Synopsys. The area overhead and impact on the functional timing due to inclusion of LTG cells is negligible.

Inserting LTG cells within the scan chain has the following advantages.

1) It increases the scan-chain controllability.
2) LTG cells can be easily tested using flush patterns during testing scan cells.
3) LTG cells can be used for both LOS and LOC methods.
4) The proposed method can be practiced along with other techniques such as pipelined scan.

VI. CONCLUSION

In this paper, a new method has been proposed to enable the design teams to practice LOS transition delay testing using low-cost testers. LOS testing is known to provide better quality results, both in terms of pattern count and fault coverage, but design teams may not use the LOS due to the challenge of routing the SEN signal. Our solution is to generate LSEN signals that can switch at functional speeds; for this purpose, we rely on embedding some control information in the patterns. We use a special cell called the LTG cell for the generation of the LSEN signal. This cell is simple to design and layout, and its area overhead is comparable to that of a scan flip–flop. We also have provided full analysis for finding the optimum number of LTG cells required. The LTG-based solution improves testability of the added hardware and provides greater flexibility and fewer constraints to the back-end flow during PNR step. The DFT insertion and ATPG can be easily performed using the commercial ATPG tools; therefore, our solution is easy to practice.

REFERENCES

[1] International Technology Roadmap for Semiconductors, 2001. [Online]. Available: http://public.itrs.net
[2] S. Natarajan, M. A. Breuer, and S. K. Gupta, "Process variations and their impact on circuit operation," in *Proc. IEEE Int. Symp. Defect and Fault Tolerance VLSI Syst.*, 1998, pp. 73–81.
[3] R. Wilson, *Delay-fault testing mandatory, author claims*. EE Design, Dec. 2002.
[4] G. Aldrich and B. Cory, "Improving test quality and reducing escapes," in *Proc. Fabless Forum, Fabless Semicond. Assoc.*, 2003, pp. 34–35.
[5] X. Lin, R. Press, J. Rajski, P. Reuter, T. Rinderknecht, B. Swanson, and N. Tamarapalli, "High-frequency, at-speed scan testing," *IEEE Des. Test Comput.*, vol. 20, no. 5, pp. 17–25, Sep./Oct. 2003.
[6] K. Cheng, "Transition fault testing for sequential circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 12, pp. 1971–1983, Dec. 1993.
[7] T. M. Mak, A. Krstic, K. Cheng, and L. Wang, "New challenges in delay testing of nanometer, multigigahertz designs," *IEEE Des. Test Comput.*, vol. 21, no. 3, pp. 241–248, May/Jun. 2004.
[8] M. Bushnell and V. Agrawal, *Essentials of Electronics Testing*. Norwell, MA: Kluwer, 2000.
[9] J. Savir, "Skewed-load transition test: Part I, Calculus," in *Proc. ITC*, 1992, pp. 705–713.
[10] J. Savir and S. Patil, "On broad-side delay test," in *Proc. VTS*, 1994, pp. 284–290.
[11] B. Dervisoglu and G. Stong, "Design for testability: Using scanpath techniques for path-delay test and measurement," in *Proc. ITC*, 1991, pp. 365–374.
[12] J. Saxena, K. M. Butler, J. Gatt, R. Raghuraman, S. P. Kumar, S. Basu, D. J. Campbell, and J. Berech, "Scan-based transition fault testing-implementation and low cost test challenges," in *Proc. ITC*, Oct. 2002, pp. 1120–1129.
[13] M. Chen Chi and S. Huang, "A reliable clock tree design methodology for ASIC designs," in *Proc. ISQED*, 2000, pp. 269–274.
[14] 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. DATE*, 2004, pp. 1296–1301.
[15] "Synopsys application note," *Tutorial on Pipelining Scan Enables*, 2004.
[16] S. Bhunia, H. Mahmoodi, A. Raychowdhury, and K. Roy, "A novel low-overhead delay testing technique for arbitrary two-pattern test application," in *Proc. DATE*, 2005, pp. 1136–1141.
[17] M. Abadir and J. Zhu, "Transition test generation using replicate-and-reduce transform for scan-based designs," in *Proc. VTS*, 2003, pp. 22–27.
[18] Y. Shao, I. Pomeranz, and S. M. Reddy, "On generating high quality tests for transition faults," in *Proc. ATS*, 2002, pp. 1–8.
[19] X. Liu and M. Hsiao, "Constrained ATPG for broadside transition testing," in *Proc. IEEE Int. Symp. DFT VLSI Syst.*, 2003, pp. 175–182.
[20] K. Yang, K. T. Cheng, and L. Wang, "TranGen: A SAT-based ATPG for path-oriented transition faults," in *Proc. ASP-DAC*, 2004, pp. 92–97.

[21] J. Rearick, "Too much delay fault coverage is a bad thing," in *Proc. IEEE ITC*, 2001, pp. 624–633.
[22] "Tetramax ATPG users guide," *SYNOPSYS Toolset Version 2004.06*, Synopsys, Inc., Mountain View, CA, 2004.
[23] "Synopsys DFT compiler," *User Manual for SYNOPSYS Toolset Version 2004.06*, Synopsys, Inc., Mountain View, CA, 2004.
[24] "Synopsys SoCBIST DBIST users guide," *User Manual for SYNOPSYS Toolset Version 2004.06*, Synopsys, Inc., Mountain View, CA, 2004.

Nisar Ahmed (S'05) received the B.E. degree in electronics and communication engineering from Osmania University, India, the M.S. degree in electrical engineering from University of Texas at Dallas, Richardson. He is currently working toward the Ph.D. degree in electrical and computer engineering at University of Connecticut, Storrs.

His current research interests include computer-aided design and test, at-speed low-cost test, and IR-drop effects on delay test.

Mohammad Tehranipoor (S'02–M'04) received the B.Sc. degree from Amirkabir University of Technology (Tehran Polytechnic University), Tehran, Iran, in 1997, the M.Sc. degree from the University of Tehran, Tehran, in 2000, and the Ph.D. degree from the University of Texas at Dallas, Richardson, in 2004, all in electrical engineering.

He was on the faculty of the Computer Science and Electrical Engineering Department, University of Maryland Baltimore County (UMBC), Baltimore, from 2004 to 2006. He is currently an Assistant Professor of electrical and computer engineering at the University of Connecticut, Storrs. He has published over 60 journal articles and refereed conference papers in the area of VLSI design and test. His current research projects include computer-aided design and test for CMOS VLSI designs and emerging nanoscale devices, design-for-testability, at-speed test, and secure design.

Dr. Tehranipoor is a member of the IEEE Computer Society, the Association for Computing Machinery (ACM), and ACM SIGDA. He has served on the program committee of several workshops and conferences in addition to serving as session chair in many technical events. He is a Guest Editor of *Journal of Electronic Testing: Theory and Applications* (JETTA) on "Test and Defect Tolerance for Nanoscale Devices" and Co-Guest Editor of IEEE Design and Test of Computers on "IR-Drop and Power Supply Noise Effects on Very Deep-Submicron Designs." He will be serving as Co-Program Chair of the 2008 International Defect and Fault Tolerance in VLSI Systems (DFT) and Program Chair of the 2007 IEEE Workshop on Defect-Based Testing (DBT). He was the recipient of a Best Paper Award at the 2005 VLSI Test Symposium (VTS) and Best Paper Candidate at the 2006 Design Automation Conference (DAC).

C. P. Ravikumar (S'88–M'90–SM'03) received the B.E. degree in electronics (with a Gold Medal for highest scores) from Bangalore University, in 1983, the M.E. degree in computer science (with highest scores) from Indian Institute of Science, in 1987, and the Ph.D. degree from University of Southern California, in 1991.

He is a Senior Technologist with the ASIC Product Development Center at Texas Instruments India. Prior to joining TI India, he was with the Department of Electrical Engineering, Indian Institute of Technology, Delhi, as a Professor, from 1991 to 2001. He was a Visiting Faculty Member with University of Southern California, from 1995 to 1996, and spent a year with Controlnet India Pvt., Ltd., Goa as a Vice President, Training (2000–2001). He has supervised four Ph.D. theses and a large number of M.Tech. and B.Tech. theses at IIT Delhi. He has authored over 200 technical papers in international journals and conferences. He is the author of the book "*Parallel Methods for VLSI Layout*" (Ablex, 1995) and has edited the series of proceedings of the annual VLSI Design and Test Symposium, which he founded in 1998 and has organized since 1998.

Dr. Ravikumar has been the Indian Editor of the *International Journal of VLSI Design* since 1991 and a member of the Editorial Staff of *Journal of Electronic Testing* (Kluwer) since 2001. He was the Coeditor of a Special Issue of the *Journal of the Computer Society of India* on the topic of Storage Area Networks. He has served as the Technical Program Chair of the International Conference on VLSI Design (2001, 2002). He has served on the program committees of several conferences such as the International Conference on High Performance Computing and the International Conference on VLSI Design. He is a recipient of the ACM/SIGDA Student Award at the IEEE/ACM Design Automation Conference (1990), the Best Student Paper Award at the IEEE International Conference on VLSI Design (1998), and the Best Paper Award at the IEEE International Conference on VLSI Design (2001). He is a member of the IETE and a Fellow of the Indian Microelectronics Society.

Kenneth M. Butler (S'84–M'85–SM'97) received the M.S. and Ph.D. degrees in electrical engineering from the University of Texas, in 1987 and 1990, respectively.

He is a TI fellow with the Quality and Reliability Engineering Department within the Silicon Technology Development Group of Texas Instruments, Dallas. In 1991, he rejoined Texas Instruments, first with the Design Automation Division and then with the ASIC Division. He spent more than 13 years working in the areas of design for testability, test-methodology development, and automated diagnosis before moving to SiTD to work on advanced test data analysis and statistical outlier identification techniques. His research interests include automatic test pattern generation and advanced applications for scan design and other structured design and test techniques.

Dr. Butler was the Program Chair of the 2005 IEEE International Test Conference. He was the corecipient of the Best Paper Award for the 1999 IEEE VLSI Test Symposium and was twice selected as the best speaker at the TI Symposium on Test.