At-Speed Transition Fault Testing With Low Speed Scan Enable

Nisar Ahmed, C. P. Ravikumar
ASIC Product Development Center
Texas Instruments India
Bangalore - 560 093
{n-ahmed2,ravikumar}@ti.com

Mohammad Tehranipoor, Jim Plusquellic
Dept. of Computer Science and Electrical Eng.
University of Maryland Baltimore County
Baltimore, MD 21250
{tehrani,plusquel}@umbc.edu

Abstract

With today's design size in millions of gates and working frequency in gigahertz range, at-speed test is crucial. The launch-off-shift method has several advantages over the launch-off-capture but imposes strict requirements on transition fault testing due to at-speed scan enable signal. A novel scan-based at-speed test is proposed which generates multiple local fast scan enable signals. The scan enable control information is encapsulated in the test data and transferred during the scan operation. A new scan cell, referred to as last transition generator (LTG), is inserted in the scan chains to generate the fast local scan enable signal. The proposed technique is robust, practice-oriented and suitable for use in an industrial flow.

I. Introduction

The semiconductor industry is adopting new fabrication processes to 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 deep-submicron (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 fault test alone cannot ensure high quality level of chips. 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].

The transition fault and path delay fault testing together provide a relatively good coverage for delay-induced defects [6] [7]. Path delay model targets the cumulative delay through the entire list of gates in a pre-defined 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]. 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 [5]. It also provides high controllability and observability.

To perform a transition fault test, a pattern pair $(V1,V2)$ is 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 $V2$ must be captured at functional speed (rated clock period). The whole operation can be divided into 3 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.

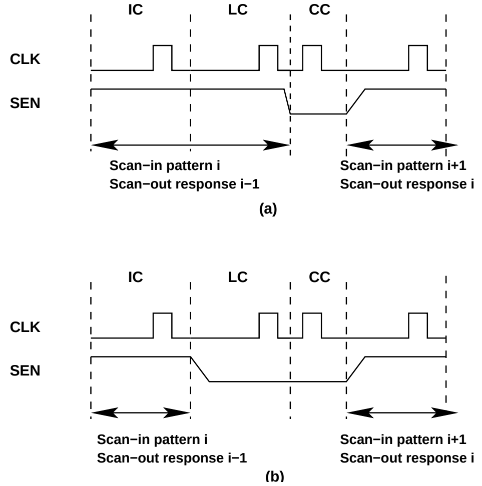

Figure 1. Transition Delay Fault Pattern Generation Methods: (a) Launch-off-shift and (b) Launch-off-capture.

Depending on how the transition is launched and captured, there are two transition fault pattern generation methods. In the first method, referred to as *launch-off-shift* (LOS), the transition at the gate output is launched in the last shift cycle during the shift operation. Figure 1(a) shows the launch-off-shift method waveform for a multiplexed-DFF design; similar approach can be applied to an 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 response capture at the CC clock egde. The time period for SEN to make this 1 → 0 transition corresponds to the functional frequency. Hence, 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. [9] list more launch and capture waveforms used by launch-off-shift approaches.

Figure 1(b) shows the waveforms of the second approach, referred to as *launch-off-capture* (LOC) method. In this method, the launch cycle is separated from the shift operation. At the end of scan-in (shift mode), pattern $V1$ is applied and CUT is set to an initialized state. A pair of at-speed clock pulses are 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 are added after the last shift to provide enough time for the SEN signal to settle low. The launch-off-shift method is more preferable based on ATPG complexity and pattern count compared to launch-off-capture method. In case of LOC, a high fault coverage cannot be guaranteed due to the correlation between the two patterns, $V1$ and $V2$.

As the design size increases, the SEN fanout exceeds any other net. One possible solution is to design the scan enable as a clock tree network but this is rarely followed due to high design cost [10]. Multiple SEN ports can be used to reduce the fanout but results in increasing the number of pins [9]. This may be a limitation for designs to be tested using very low cost testers (VLCT). In [11], a hybrid architecture is proposed which controls a small subset of selected scan cells by launch-off-shift and the rest are controlled by launch-off-capture approach. A fast scan enable signal generator is designed which drives the LOS controlled scan flops. The ATPG method used is complex and current commercial tools do not support such a technique. Moreover, the selection criteria of the LOS controlled scan flops determines the effectiveness of the method. In some cases the number of patterns generated by the hybrid method exceeds the launch-off-capture patterns.

A very widely used method is to pipeline the scan enable signal [12]. In a multi-stage pipeline scan enable, the designer must carefully select the group of scan cells controlled by the respective scan enable signal. In order to meet timing closure of the pipeline scan enable signals, this selection criteria manifests into design iterations and additional constraints for the place and route (PNR) tool.

In this paper, we propose a scheme where the scan enable control information for the launch and capture cycle is embedded in the test data itself. A new scan cell, called the last transition generator (LTG), generates the local fast scan enable signals. The LTG cell has the flexibility to be inserted anywhere in the scan chain and the hardware area overhead is comparable to the pipeline scan enable approach. The proposed method poses no additional constraints for the place and route tool and provides more flexibility to re-order the scan cells to meet the timing closure of the local scan enable signals.

The rest of the paper is organized as follows. The pipeline scan enable method is reviewed in Section II. Section III describes the local scan enable generation and the architecture and operation of LTG cell. Section IV explains the test architecture and launch-off-shift clock timing waveforms. The DFT insertion and test protocol file changes for ATPG along with the experimental results are discussed in Section V. Finally, concluding remarks are in Section VI.

II. Pipeline Scan Enable

The launch-off-shift method requires the scan enable signal to be at-speed. Figure 2 shows the basic implementation and operation of a pipelined scan enable for launch-off-shift [12]. The scan enable port (*SEN_port*) is de-asserted asynchronously in the initialization cycle and the pipeline scan enable (*SEN_pipeline*) goes low synchronously at the active edge of launch clock cycle. The *SEN_port* is asserted high after the capture cycle to re-enter the shift mode. The pipeline scan enable architecture provides two significant advantages. Firstly, the *SEN_pipeline* signal is generated synchronously and has a full clock cycle available for transition, assuming the pipeline scan enable signal is generated separately for positive and negative edge flops. Secondly, the tester skew at the scan enable port is decoupled from the internal scan enable net.

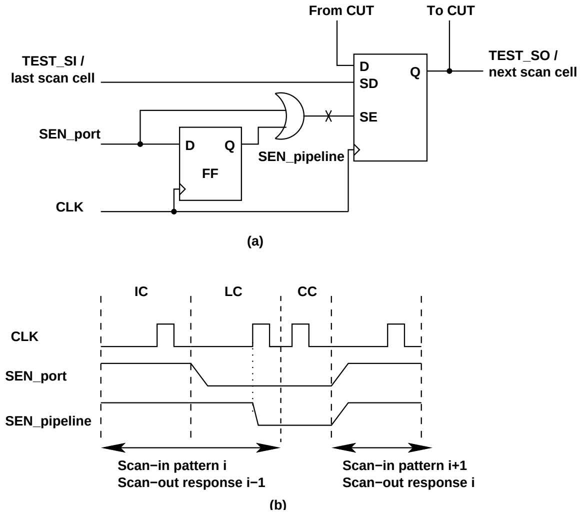

Figure 2. Pipeline Scan Enable.

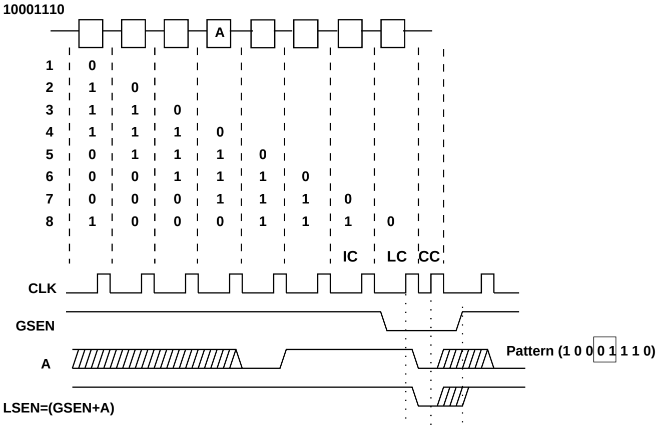

Figure 3. Generation of local scan enable signal.

The pipeline scan enable method requires routing awareness before insertion. The number of pipelined flops added in the scan enable path depends on the worst case delay. Multiple pipeline scan enable stages can be added depending on the distance of the last stage driving the scan enable network. However, multiple stage pipelines have increased susceptibility to noise glitches from tester or due to crosstalk. An assertion glitch on the first pipeline stage would force scan enable active for multiple clock cycles. Note that, the pipeline scan enable scheme is an industry practiced technique and the proposed technique discussed in the next section (Section III) provides more robustness and can be practiced along with the pipeline scan enable scheme.

III. Local Scan Enable Signal Generation

A. Motivation

The pipeline scan enable methodology provides a good mechanism to divide the fanout of the scan enable signal without using multiple external pins and eliminates the external scan enable port tester skew. It is independent of the floor planning step and does not provide much flexibility to the PNR tool in terms of selecting the scan cell group driven by the pipeline scan enable, which in some cases may result in design iterations. In order to provide more flexibility with all the advantages of the pipeline scan enable, the local scan enable generator cells are inserted within the scan chains. Therefore, the control information is to be passed as part of the test data. The new architecture provides the PNR tool with more flexibility to reorder the scan chain cells including the scan enable generator cells during timing closure step.

Figure 3 shows a small example of generating the local scan enable signal from the test pattern data for LOS. The external scan enable signal from the tester is referred to as the global scan enable (GSEN). The internally generated scan enable signal is termed as local scan enable (LSEN). The main objective is to de-assert GSEN in the initialization cycle (not at-speed) and then generate the LSEN signal during the launch and capture cycle synchronously from the test data. There are eight scan flops in the scan chain and the test pattern shifted is *(10001110)*. The values of the scan flops during the various shift cycles are shown under each flop. GSEN is de-asserted during the $(n-1)$th shift (IC) cycle, 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 flop *A*. The output of scan flop *A* is ORed with GSEN to generate the LSEN signal. Therefore, the final value of scan flop (*A*) and its following scan flop at the end of shift operation must be 0 and 1, respectively, so that *A* is loaded with logic 1 in IC and logic 0 in LC. *A* full at-speed cycle is available for LSEN to make the transition. After the capture cycle, the LSEN signal is aysnchronously set to 1 by GSEN for scanning out the response.

B. Last Transition Generator (LTG)

As explained earlier, during launch-off-shift pattern generation, to generate the scan enable transition $1 \rightarrow 0$ at the launch edge, the scan flop *A* should be 0 and the following scan flop be 1. This is very unlikely to occur in every pattern generated during atpg. It can also be seen in Figure 3 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 LOC pattern generation. For LOC patterns, the LSEN signal must be de-asserted and two system clocks are applied. Hence, *A* must be atpg constrained to 0 during LOC pattern generation. This constraint is necessary but not enough 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), 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 flop architecture to control the functional path (D) to logic 0. This might lead to fault coverage reduction.

In order to avoid loss of coverage without significant change in the architecture, a new scan cell called *last transition generator* (LTG) is designed such that when the scan cell is loaded with logic 0 during shift and the scan enable control is made logic 0, i.e. capture state, the cell will remain in that state until it is shift controlled to 1. This cell is inserted into scan chain to generate the $1 \rightarrow 0$ transition at the launch edge. Figure 4 shows the LTG cell architecture. It consists of two flops which are used to load the control information required for the launch and capture cycles. The port definition is similar to a scan cell and the output of FF1 is fed back to the functional input port of the scan cell. The input *D* actually does not exist because LTG cell does not connect to CUT. We represented here to compare LTG cell with normal scan cell. It consists of a scan-in (SEN_{in}) pin which takes GSEN signal as input. An additional scan-out (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.

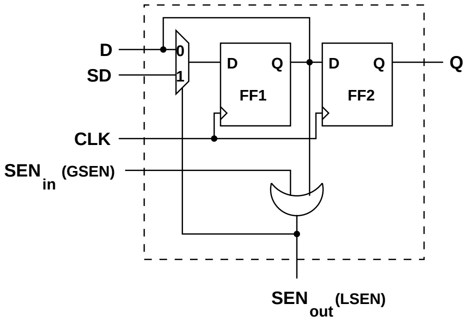

Figure 4. Last transition generator (LTG) cell.

Theorem: The local scan enable signal generated by the LTG cell switches at-speed during the capture cycle.

Proof: SEN_{out} refers to the local scan enable signal in the LTG cell of Figure 4. The clock input to the LTG cell for launch-off-shift transition delay ATPG is of the form shown in Figure 3. 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. During the scan shift cycle (IC), a ”1” is scanned into the 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 flops driven by SEN_{out} is an order of magnitude smaller than the total number of flops in the design, thereby reducing the capacitive load on the local scan enable signal. The global scan enable signal switches to 0 during the beginning of the LC cycle. Let A refer to the output of FF1. SEN_{out} is the logical OR of the signal A and the global scan enable signal; therefore, SEN_{out} also switches at the speed of signal A, except for the small delay in the OR gate.

C. Operation of LTG cell

Figure 5(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 flops of the LTG cell in the test pattern are shown as X (1000[XX]1110). These flops are constrained during atpg to assign specific values to the Xs. Figure 5(b) shows the pattern and the timing waveform for LOS. During the shift operation, at the last shift the scan enable 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 SEN_{in} (GSEN) signal is asynchronously deasserted in IC. The SEN_{out} (LSEN) signal is generated by the boolean equation

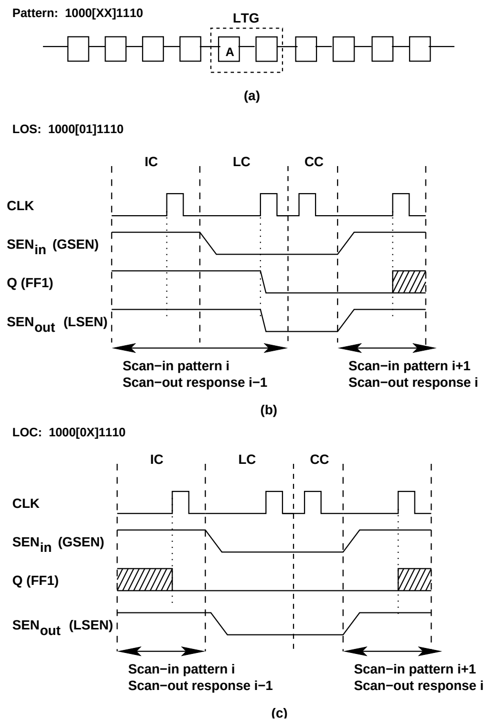

Figure 5. Operation of LTG cell, (a) Example scan chain, (b) LOS and (b) LOC.

$SEN_{out} = Q(FF1) + SEN_{in}$. After the capture cycle, the LSEN is asserted to 1 asynchronously by GSEN.

For LOC, the GSEN signal is high during the entire shift operation. At the end of the shift operation, the GSEN signal is asynchronously deasserted and the LSEN signal must be logic 0. The value of FF2 does not affect the operation, i.e. FF2 is used for storing the output of FF1 during last shift. Since LTG cell is not connected to the CUT, therefore the content of FF1 and FF2 are not part of applied test pattern to CUT. The value of FF1 of LTG cell must be constrained to 0 during atpg. Figure 5(c) shows the pattern and the timing waveform for LOC. The LSEN signal is asynchronously de-asserted and asserted back by the GSEN signal. It can be noticed that these transitions are not at-speed.

Comparing LOS timing waveforms of Figures 5(b) and 1 shows that in Figure 1 since the same scan enable feeds both positive and negative edge flops, it has to make a transition after the negative edge of LC. As a result, the time for it to go low is half the clock cycle. While in our method the local scan enable signal is generated separately for negative edge and positive edge flops. This gives a complete cycle for the scan enable signal to make a transition (see Figure 5(b)).

IV. Test Architecture

The LTG-enabled solution presented in this paper considerably eases the problem of routing the scan enable signal by taking in a global scan enable signal that need not switch at functional speed and generating the local scan enable signals internally. The number of local scan enable signals can be specified by the user. The overhead of generating the local scan enable signal is the addition of an LTG cell in the scan chain. The area overhead of an LTG cell is a few extra gates, which is negligible in modern designs. Each local scan enable signal drives a fraction of the total number of flip-flops in a clock domain. The question naturally arises on what is the largest number of LTG cells one can insert. One can look at the distribution of flops in the layout of the chip (see Figure 6).

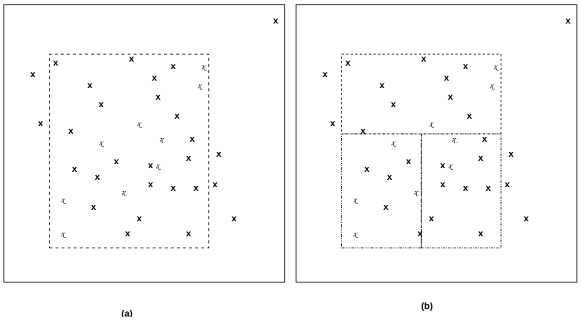

Figure 6. (a) Distribution of fbps in the chip layout. (b) Partitioning the set of fbps for predictable closure on scan enable timing.

The difficulty of timing closure for the scan enable signal can be estimated by some function *f* of the area of smallest bounding rectangle that covers the majority of the flops (dotted rectangle in Figure 6(a)). The rectangle then needs to be partitioned such that we have nearly equal number of flops in each of the partitions, and each bounding sub-rectangle is small enough to limit the function *f* to an acceptable value as shown in Figure 6(b). The process of partitioning can be performed through the repeated use of a bi-partitioning algorithm such as Kernighan-Lin [13]. Note that the assignment of flops to scan chains ordering of the flops within the scan chains is not directly relevant to the partitioning problem; therefore, this step can be performed after the physical design flow has reordered the flops in the scan chain and optimized the placement and routing for functional timing closure. The location of the LTG cell in a scan chain can be selected such that the cell is nearly in the center of the leaf-level rectangle.

In general, there can be multiple scan chains in a design to reduce the test application time. The test application time is directly proportional to the longest scan chain length in the design. Figure 7 shows a multiple scan chain architecture with n scan chains. Each scan chain i, where $1 \leq i \leq n$, consists of a LTG cell which generates the fast scan enable signal $LSEN_i$. Note that, if scan enable 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. The fanout load on the global scan enable (GSEN) signal is reduced and the fanout load driven by a local scan enable signal is used as a constraint to find the number of LTG cells inserted. For example, for m total number of flops in a design and n being the maximum number of flops that can be allowed for the local scan enable to be timing closed for a particular operating frequency, the number of LTG cells are estimated by m/n.

The methodology is not effected by multiple clock domains. Since all scan chains are shifted at the same speed, the launch edge of all the clock domains occur at the same time. The timing constraint for the LSEN signal is that it must be timing closed for the fastest clock domain in the scan chain. 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. Figure 8(a) shows the limitation of the clock timing waveform. The functional operating frequency is

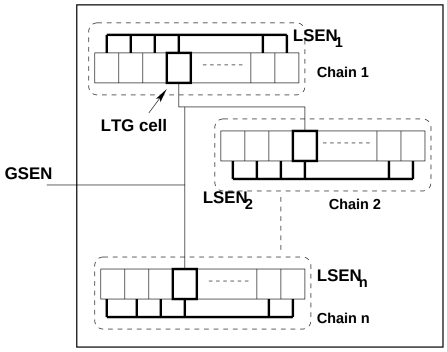

Figure 7. Test Architecture.

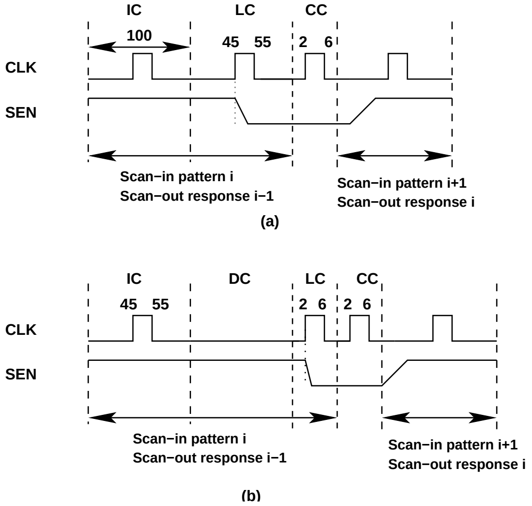

Figure 8. LOS clock timing waveform.

125MHz. The launch edge in the last shift occurs at 45ns and the capture edge occurs at 2ns in the capture cycle of 8ns time period. The time from the launch edge to the capture edge is (55+2)=57ns. Figure 8(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 dead cycle (DC) is added after the initialization cycle for the scan chain to settle down.

V. Experimental Results

In this paper, we have argued in favor of the 'launch off shift' transition delay ATPG methodology and presented a technique that can ease the implementation of this technique. We experimented with an industrial-strength design that had the following characteristics (Table I). The design has 16 scan chians and almost 70K scan cells. There are 97 non-scan cells and two internal clock domains, 125MHz and 250MHz, respectively. The test strategy is to get the highest possible test coverage for the stuck-at faults. When generating test patterns for the transition faults, we target only the faults in the same clock domain. During pattern generation, only one clock is made active during the capture cycle. 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 results for LOS and LOC transition-delay ATPG on this design are shown in the Table II. TetraMAX [14] tool was used for ATPG. We see that LOS methodology gives about 4% higher coverage than the LOC methodology. In separate experiments performed at our organization, it has been independently confirmed that LOS technique gives up to 10% higher coverage on most designs. For the design under consideration, we found that using 1312 patterns, the LOS method gave the same coverage as the LOC method. This represents a reduction of about 88% as compared to LOC. The main barrier to the practice of LOS is the difficulty in closing the timing on the scan enable signal, as a result, LOS is not used on designs where the turn-around-time is critical. The price to be paid for this is two fold - (a) increase in test cost, since the pattern volume for LOC is higher, and (b) reduction in coverage, which impacts the *defective parts per million* (DPPM) metric.

TABLE I
Design Characteristics

Transition Delay Faults	4363144
Scan Flops	69836
Non-scan Flops	97
Scan Chains	16
Clock Domains	2

TABLE II
Experimental Results

	Launch-off-Capture (LOC)		*Launch-off-shift (LOS)*	
	*CLK*1	*CLK*2	*CLK*1	*CLK*2
Detected faults	466687	3733728	466707	3893150
Test Coverage	10.7	85.75	10.7	89.44
Pattern Count	3	9401	3	14332
CPU Time *[sec]*	235792		172031	
Memory Usage *[MB]*	1031		1194	

A. DFT Insertion

Synopsys DFTCompiler [15] is used for scan chain insertion in a design. Figure 9 shows the list of additional commands in the *tcl* script. Here, we assume that one LTG cell is inserted per scan chain. To insert the LTG cells, additional commands are required during the scan chain configuration. The synthesis tool must recognize the LTG cell as a scan cell in order to stitch it into the scan chain. This requires it to be defined as a new library cell with the scan cell attributes. A workaround is to design the LTG cell as a module and declare it as a scan segment using the *set_scan_segment* command (*line 04*). The tool then identifies LTG cell as a scan segment of length 2. The GSEN signal is connected to all the LTG cells SEN_{in} input pin. To make the *insert_scan* command insert the LTG cells in the scan chain, *set_scan_path* command must be used to declare the scan path (*line 09*). Only the LTG cell is specified in the scan path, as the tool will stitch the rest of the cells including the LTG cell and balances the scan chain depending on the longest scan chain length parameter defined in the *set_scan_configuration* command. The *set_scan_signal* command (*line 10*) is used to hookup each LTG cells SEN_{out} port in a particular chain to all the scan enable input port of the scan flops in the respective chain.

B. ATPG

There is no fundamental difference in the ATPG methodology when we use the LTG-based solution. The scan enable sig-

```
for {set i 0} {$i<no_chains} {incr i}
{
    create_cell LTG$i LTG
    set_scan_segment scan_segment$i
        -access {test_scan_in LTG$i/SD, test_scan_out LTG$i/Q}
        -contains {LTG$i/FF1, LTG$i/FF2}
    connect_net GSEN find( pin, LTG$i/SEN_IN)
    connect_net CK find( pin, LTG$i/CLK)
    set_scan_path c$i LTG$i -dedicated_scan_out true -clock CK
    set_scan_signal test_scan_enable -port GSEN
        -hookup find( pin, LTG$i/SEN_OUT) -chain c$i
}
```

Figure 9. DFTCompiler Tcl Script commands.

```
”load_unload”{
    W ”_slow_WFT_”;
    V { ”CLK1”=0; ”CLK2”=0;”GSEN”=1;}
    Shift {
        W ”_slow_WFT_”;
        V { ”CLK1”=P; ”CLK2”=P; ”GSEN”=1; ” so_”=#; ” si_”=#; }
    }
    //ADDING DEAD CLOCK CYCLE
    V { ”CLK1”=0; ”CLK2”=0; ”GSEN”=0; }
    W ”_fast_WFT_”;//Nth SHIFT CYCLE
    V { ”CLK1”=P; ”CLK2”=P; ”GSEN”=0; ” so_”=#; ” si_”=#; }
}
```

Figure 10. TetraMAX ATPG protocol file.

nal for the flops now comes from an internally generated signal. An ATPG tool must be able to place the scan enable signal in the active mode during scan shift operation. In order to get around the problem of the ”design rule check” phase of the ATPG tool failing due to the internal nature of the scan enable signals of the flops, we set the global scan enable GSEN signal to 1 during test setup. Notice that the OR gate in the LTG cell generates the local scan enable signal through a logical OR of the global scan enable and the Q output of the flop FF1 (see Figure 4). As a result, the DRC phase of the ATPG passes without any violations. In a commercial ATPG tool, the loss in coverage due to undetectability of stuck-at-1 faults on the scan enable signals can be recovered by declaring the scan enable signal as a clock signal. Figure 10 shows the *load_unload* procedure of the test protocol file. Each vector (*V*) statement is a tester clock cycle. The waveform table (*W*) statement determines the clock period of the tester clock cycle defined in the timing waveform procedure of the test protocol file. The waveform table $\_slow\_WFT\_$ is the slow shift clock cycle and $\_fast\_WFT\_$ is the at-speed clock cycle. The GSEN signal is high till the (n-1)th shift and is made low in the dead cycle before the last shift which provides enough time for GSEN to go low. The last shift is done at-speed and it can be seen that the waveform timing is changed to $\_fast\_WFT\_$ before the last *V* statement.

We believe that the LTG-based solution has the following advantages. The technique can be practiced using existing commercial tools for DFT insertion and ATPG. The technique is applicable to all scan-based DFT techniques, including Deterministic BIST techniques that are based on scan [16]. The method is least intrusive and can be incorporated in existing physical design flows. 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. Finally, the 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 launch-off-shift (LOS) transition delay testing. LOS testing is known to provide better quality results, both in terms of pattern count and fault coverage, but design teams may not use launch-off-shift due to the challenge of routing the scan enable signal. Our solution is to generate local scan-enable 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 local scan enable signal. This cell is simple to design and layout, and its area overhead is comparable to that of a scan flop. The number of LTG cells inserted in the design will be small, thereby making the area overhead due to our technique negligible. The LTG-based solution provides greater flexibility and fewer contraints to the backend flow during place and route 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 (http://public.itrs.net).

[2] S. Natarajan, M.A. Breuer, S.K. Gupta, “Process Variations and Their Impact on Circuit Operation,” in Proc. *IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, pp. 73-81, 1998.

[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 Semiconductor Assoc.*, pp. 34-35, 2003.

[5] X. Lin, R. Press, J. Rajski, P. Reuter, T. Rinderknecht, B. Swanson and N. Tamarapalli, “High-Frequency, At-Speed Scan Testing,” *IEEE Design & Test of Computers*, pp. 17-25, Sep-Oct 2003.

[6] K. Cheng, “Transition Fault Testing for Sequential Circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 12, pp. 1971-1983, Dec. 1993.

[7] T. M. Mak, A. Krstic, K. Cheng, L. Wang, “New challenges in delay testing of nanometer, multigigahertz designs,” *IEEE Design & Test of Computers*, pp. 241-248, May-Jun 2004.

[8] M. Bushnell, V. Agrawal, *Essentials of Electronics Testing*, Kluwer Publishers, 2000.

[9] J. Saxena, K. M. Butler, J. Gatt, R. Raghuraman, S. P. Kumar, S. Basu, D. J. Campbell, J. Berech, “Scan-Based Transition Fault Testing - Implementation and Low Cost Test Challenges,” in Proc. *International Test Conference (ITC’02)*, pp. 1120 - 1129, Oct. 2002.

[10] M. Chen Chi; S. Huang; “A reliable clock tree design methodology for ASIC designs,” in Proc. *International Symposium on Quality Electronic Design (ISQED’00)*, pp. 269-274, 2000.

[11] S. Wang, X. Liu, 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. *Design, Automation and Test in Europe (DATE’03)*, pp. 1296-1301, 2004.

[12] ———, “Tutorial on Pipelining Scan Enables,” *Internal Document*.

[13] B.W. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,” *Bell System Tech. Journal*, vol. 49, pp. 291-307, 1970.

[14] Tetramax ATPG Users Guide, “SYNOPSYS Toolset Version 2004.06,” Synopsys, Inc., 2004.

[15] Synopsys DFT Compiler, “User Manual for SYNOPSYS Toolset Version 2004.06,” Synopsys, Inc., 2004.

[16] Synopsys SoCBIST DBIST Users Guide, “User Manual for SYNOPSYS Toolset Version 2004.06,” Synopsys, Inc., 2004.