Timing-Based Delay Test for Screening Small Delay Defects

Nisar Ahmed, Mohammad Tehranipoor
Dept. of Computer Engineering
Univ. of Maryland Baltimore County
{nisar1, tehrani}@umbc.edu

Vinay Jayaram
Texas Instruments, Inc.
Dallas, TX
vjayaram@ti.com

Abstract

The delay fault test pattern set generated by timing unaware commercial ATPG tools mostly affects very short paths, thereby increasing the escape chance of smaller delay defects. These small delay defects might be activated on longer paths during functional operation and cause a timing failure. This paper presents an improved pattern generation technique for transition fault model, which provides a higher coverage of small delay defect that lie along the long paths, using a commercial no-timing ATPG tool. The proposed technique pre-processes the scan flip-flops based on their least slack path and the detectable delay defect size. A new delay defect size metric based on the affected path length and required increase in test frequency is developed. We then perform pattern generation and apply a novel pattern selection technique to screen test patterns affecting longer paths. Using this technique will provide the opportunity of using existing timing unaware ATPG tools as slack based ATPG. The resulting pattern set improves the defect screening capability of small delay defects.

Categories and Subject Descriptors: B.8.1 [Integrated Circuits]: Performance and Reliability-Reliability, Testing and Fault-Tolerance

General Terms: Reliability.

Keywords: delay testing, test generation.

1. INTRODUCTION

Scan-based structural tests are increasingly used as a cost-effective alternative to the at-speed functional pattern approach [1] [2]. Transition fault and path delay fault are two prevalent fault models and together provide a relatively good fault coverage for delay-induced defects [3] [4]. The transition fault model targets each gate output in the design for a *slow-to-rise* and *slow-to-fall* delay fault while the path delay model targets the cumulative delay through the entire list of gates in a pre-defined path [5].

Transition fault model is widely practiced in industry and existing commercial tools have matured in test generation and debug of such tests. The traditional transition fault tests are generated assuming a fixed cycle time for each clock domain (generally delay tests are generated/applied one clock domain at a time). This shows that a delay defect will be detected only when it causes a transition to reach an observe point (primary output or scan flip-flop) by more than the positive slack of the affected path. Slack of a path is a measure of how close a transition on the respective path meets the timing of an observe point, relative to the test cycle time.

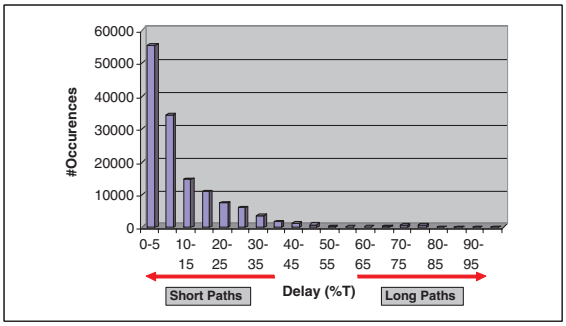

Figure 1: Path delay distribution for no-timing ATPG transition fault pattern set (benchmark s38584).

A small delay defect might escape, if activated through a short path during test. While the same defect might be activated on a long path during functional operation and it may cause a timing failure. The detection of small delay defects on long paths is a quality issue. For a particular defect, a pattern which affects a longer path is more efficient than a pattern which detects it through a shorter path. Whereas, the detection of small delay defects on short paths is more of a reliability issue. A small delay defect escape on such paths during test might magnify during subsequent aging in the field and cause a failure of the device. In this work, we focus only on small delay defects on long paths to improve the quality of delay test pattern set. Moreover, the definition of long paths is very important and it depends on the frequency. If the frequency is increased, the slack of the path decreases and hence more paths which were not considered long enough earlier will become timing critical.

There is a growing industry concern for timing aware ATPG tools. Encounter True-Time Delay Test Tool ™[6] is one such available commercial ATPG tool that uses actual design timing information for ATPG. However, most of the widely used commercial ATPG tools are still timing unaware and generate test patterns based on the ease of finding an affected path, instead of a least slack path.

Figure 1 shows the delay distribution of a transition fault pattern set generated using a commercial timing unaware ATPG tool (Synopsys TetraMax [7]) * for an ISCAS'89 benchmark (s38584). The patterns were generated using launch-off-capture technique (*broadside*) for the total transition fault list (52874 faults) and the gross

*Results obtained by TetraMax did not consider any slack based options.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

delay fault coverage and pattern count were 76.92% and 372 respectively. It can be noticed that majority of the paths exercised for delay fault detection are short paths. The minimum defect size detectable depends on the path delay region affected by the pattern set, relative to the clock cycle. For instance, in this particular example, most of the paths affected are less than 30% of cycle time. A delay defect size of at least 70% cycle time is required for the faults to be detected. Therefore, more robust at-speed techniques are required to improve the effectiveness of transition fault testing to affect more longer paths and screen the small delay defects better.

1.1 Related Prior Work

Various techniques have been proposed in the past for improving the small delay defect screening quality of a pattern set. A number of these methods such as very-low-voltage (VLV) [8] and burn-in [9], modify the operating conditions of test environment and magnifies the defect size, which escape at nominal conditions. However, in DSM designs, the effectiveness of VLV testing is reducing as i) the scaling of threshold voltage is not proportionate to supply voltage and ii) issues like IR-drop and crosstalk are becoming more prominent and burn-in is associated with considerable high costs.

Alternative methods are under investigation to detect such defects. In [10], a transition fault model, called As Late As Possible Transition Fault (ALAPTF) was proposed. The method tries to activate and propagate a transition fault at the target gate terminal through the least slack path possible. The ATPG method used is complex and will be more CPU intensive compared to a no-timing ATPG. A delay fault coverage metric is proposed in [11] which tries to detect the longest path affecting a line. The technique attempts to find the longest sensitizable path passing through the target line producing a rising (falling) transition on it. In [12], the authors proposed a new ATPG tool to generate *K* longest paths per gate for transition fault test. The technique targets all the transition faults to find the longest path. A longest path does not reflect the detectable delay defect size. For example, if the least slack path of a gate is a short path then a small delay defect on such a gate output cannot be detected for the nominal frequency.

The technique proposed in [13] is based on detecting a smaller delay on a shorter path by increasing the frequency of operation, instead of detecting it on a long path which requires a timing aware ATPG tool. Due to increasing the frequency, the capture edge might occur in the hazard region for some of the observation points. Such methods may also be limited by the highest possible frequency of operation which exacerbates the already well known issues of peak power during test and IR-drop. The authors in [14] showed a case study of the effects of IR-drop and explored quiet pattern (reduced transition) generation methods to reduce it. Recently, the effects of power supply noise on clock frequency during delay test was presented in [15].

In [16], multiple-detect test pattern sets are used to improve the quality of tests by maximizing the probability of detecting bridging defects but generates high pattern count compared to a single detect pattern set. To enhance the effectiveness of screening frequency dependent defects, the authors in [17] propose a pattern selection methodology to reduce the delay variation of the selected pattern set and higher frequency is used for pattern application. The method uses a multiple-detect transition fault pattern set and it uses statistical timing analysis techniques to reduce pattern delay variations.

1.2 Contribution and Paper Organization

The pattern modification or selection techniques described above assume that a single-detect or a multiple-detect pattern set is already available, respectively. Moreover, these methods try to improve the defect screening effectiveness by detecting small delay defects on all paths of the design at multiple higher test frequencies, which might further worsen the issues of power during test and IR-drop.

In this paper, we propose a new pattern generation technique which targets the small delay defects only on the long paths of a design for the functional operating frequency. But the user can define multiple test frequencies based on his analysis for tolerable increase in test power and IR-drop. The proposed technique uses static timing analysis tool to divide the path lengths and their corresponding observation points into different categories (long, intermediate and short paths respectively). We define a new delay defect size metric based on the affected path length category and required increase in test frequency.

We then perform multiple-detect ATPG on all the fault sites along the long paths to detect small delay defects. A novel pattern selection technique is used which selects patterns activating higher percentage of long paths and masks all short paths. Using this technique, we can use existing timing unaware ATPG tools to obtain a high coverage of small delay defects along the long paths of a design. The experimental results show the effectiveness of our proposed technique in detecting small delay defects on longer paths when compared to traditional no-timing pattern generation.

The rest of the paper is organized as follows. Section 2 explains the path length analysis and the basic idea to group the paths and observe points. The ATPG methodology used to generate patterns is discussed in Section 3. The pattern selection method is described in Section 4. The experimental results are presented in Section 5. Finally, concluding remarks are in Section 6.

2. PATH LENGTH ANALYSIS

A *gross* delay defect can be detected irrespective of the affected path. While a *very small* delay defect can be detected only when affected by an extremely timing sensitive path, referred to as a critical path. Such paths are limited in number and they are used for path delay fault test. A *small* delay defect which cannot affect any critical paths needs to be detected through their least possible slack path. There are millions of paths in a design with different paths of varying length converging to each observation point. To differentiate between all the various paths in a design, we define three categories of paths based on their path length and the minimum size of the delay defect that can be detected through the path.

1. **Long Path (LP):** A long path in a design is defined as a path, if affected by a small delay defect can cause a timing failure. Such paths are timing sensitive as the path's timing length is very close to the target frequency next to the critical path (CP) category.

2. **Short Path (SP):** A short path requires a significant delay defect size that will create a very large timing variability to cause a failure. Detecting small delay defects on such paths requires a very high test frequency, to make the slack near zero.

3. **Intermediate Path (IP):** A path with a delay in the range other than long paths and short paths is defined as an intermediate path.

The path length range of each category is shown in Figure 2. The vertical axis is organized with the least slack at the top and the highest slack at the bottom. The cutoff limits of the long and short path's region are shown as LP-cutoff and SP-cutoff, respectively.

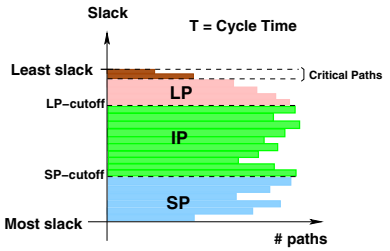

Figure 2: Different categories of paths (long, intermediate and short paths).

Table 1: Comparison of delay defect size and test frequency for different path categories.

Path Type	Defect Size Range	Frequency Increase
CP	very small - gross	No change
LP	small - gross	Slight - No change
IP	intermediate - gross	Medium
SP	gross	High

These cut-off points can be determined based on the delay defect size coverage and the increase in frequency required to make the slack of the path near zero.

Table 1 compares the detectable delay defect size at nominal functional frequency and the test frequency increase required compared to the functional frequency for each path length category to detect small delay defects. The critical paths (CP) are very timing critical and even very small delay defects can be detected on such paths. To detect small delay defects on LP paths, it requires less increase (or no increase) in test frequency. As a result, we don't expect test power or IR-drop increase. Whereas, detecting small delay defects on IP paths requires higher test frequency. The IP paths are important because as we increase the test frequency some of the paths in IP category become long paths for the next target frequency. On short paths (SP), only gross delay defects can be detected. To detect small delay defects on short paths, a very high test frequency (probably $\geq 2X$ increase) will be required.

The above path length analysis shows that small delay defects on intermediate and short paths can only be detected using higher test frequencies. In this paper we consider only the small delay defects on long paths for pattern generation. However, if increasing frequency is not an issue, our technique is flexible and it can be applied for small delay defects on all paths.

An observation point at the end of a path (primary output or scan flip-flop) is referred to as an *endpoint*. Note that, during delay testing, the primary outputs are not measured between the launch and capture cycles. This is due to insufficient timing accuracy of a low-cost tester to strobe the primary outputs before the capture event. Therefore, in the rest of the paper, an endpoint refers to a scan flip-flop.

Each endpoint is associated with a path delay distribution. Due to the complexity of finding all the paths to an endpoint, we consider only the least slack path to each endpoint. Figure 3 shows the number of endpoints with least slack (benchmark s38584) divided across the entire cycle period. A static timing analysis tool (Synopsys PrimeTime [7]) was used to find the least slack path to each endpoint. In this example, we set the LP-cutoff limit as slack less than 30% of cycle time. It can be noticed that we included the critical path endpoints in the LP category as they might contain long paths in their path distribution. Similarly, a slack greater than 70% of cycle time is considered for the SP-cutoff limit.

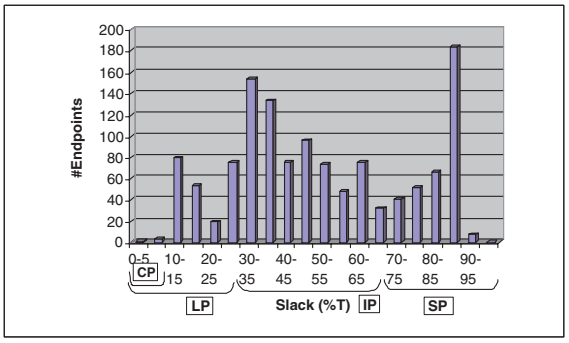

Figure 3: Slack versus endpoints (benchmark s38584).

The static timing analysis tool measures the slack of a path by $slack = T_{cycle} - T_{setup} - T_{delay}$, where T_{cycle}, T_{setup} and T_{delay} refer to cycle period, setup time and delay of path, respectively. A small percentage of endpoints (approx. 12%) have their least slack in the long path category. A majority of the endpoints with least slack fall in the intermediate category (approx. 61 %), while the remaining (approx. 27 %) fall in the short path length category.

3. ATPG METHODOLOGY

As shown in the previous section, the delay defects on the long paths require a smaller defect size to cause a timing failure compared to intermediate and short paths. Such small delay defects on long paths might escape at nominal frequency during test, since a timing unaware ATPG tool will affect short paths through them. One possible solution to detect such defects is to extract all the long paths in the design and perform path delay test pattern generation. If it could achieve 100% path delay coverage then the defect coverage of all small delay defects on long paths would be 100%. However, due to inherent robust pattern generation, path delay test gives a very small coverage of long paths. Therefore, we require heuristics to detect small delay defects on long paths using transition fault model.

Each LP endpoint will have several short and intermediate paths converging to it other than a long path. In order to force the timing unaware ATPG tool not to exercise the short paths, the initializing points of the short/intermediate paths need to be held constant. Such logic sensitization control (holding certain logic state's constant) might be very complex. In case of launch-off-capture (broadside) method which launches a transition through the functional path will require the next time-frame information to determine the present state values to hold them constant. This requires a detailed analysis to find all the initializing endpoints and hold them constant during pattern generation. This is not possible due to the very high computational cost.

To avoid the high cost and complexity of path delay analysis and still increase the probability of a fault to be detected using a long path we use multiple-detect method. Multiple-detect technique tries to activate the fault site through multiple paths. Since, most of the ATPG tools have the capability of multiple-detect pattern generation, we utilize this capability to activate the long paths. Figure 4 shows the pattern delay distribution of a 15-detect transition fault pattern set for the same benchmark. The LP-endpoints are only used as observe points and the remaining endpoints (IP and SP) are masked. This is performed using cell constraints during pattern generation which forces the ATPG tool to ignore the value captured in the respective endpoints. However, an endpoint with a cell constraint can still be loaded with a valid care-bit value during scan chain shift for transition launch and propagation.

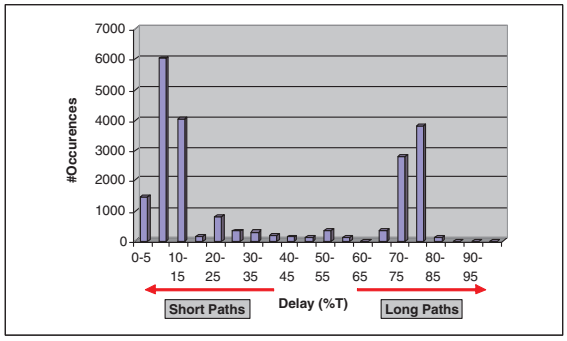

Figure 4: Path Delay distribution for 15-detect pattern set (benchmark s38584).

The histogram excludes the endpoints that do not observe a transition since they do not contribute to fault coverage. The number of long paths affected in a 15-detect pattern set is higher compared to a 1-detect pattern set (see Figure 1). However, there are still a lot of short paths being affected to the LP-endpoints. Hence, our goal is to select a subset of patterns from a 15-detect pattern set with majority of long paths used to observe the delay faults. The 15-detect pattern set is larger than a single detect pattern set due to multiple combinations of activating each fault site. For this particular experiment, approx. 1000 patterns were generated for a 15-detect compared to 300 patterns for a 1-detect delay fault pattern set.

The ATPG process is divided into three steps. In the first step, we perform path delay test to cover some of the small delay defects on the long paths. In the second step, we consider only the LP-endpoints to be observable for delay fault test generation. The remaining endpoints (IP and SP) are made non-observable. The SP endpoints are masked, since these endpoints with least slack in the SP category require a huge delay fault in order to fail. In the final step, we repeat the above multiple-detect pattern generation using only IP-endpoints as observe points. This step is optional and depends on whether we want to apply higher frequency for test. The IP path range is the next range of paths close to LP. Any increase in frequency will push the IP paths into LP range for the new target frequency. Increase in frequency creates a possibility of hazards in patterns, especially to LP-endpoints as their path range might exceed the new target clock period. In such a case, the respective endpoints are required to be masked to avoid any timing failures.

4. PATTERN SELECTION

As mentioned earlier, a multiple-detect pattern set is larger than a single detect pattern set and there are still many short paths being exercised. Therefore, we select a subset of patterns from the multiple-detect pattern set with majority of long paths being affected. To perform the selection, we analyze the path delay distribution of each pattern in the pattern set. Suppose, there are two types of patterns $P1$ and $P2$ respectively in a delay fault pattern set. The pattern $P1$ affects multiple paths in the LP range while pattern $P2$ affects only a single long path and rest are short paths. For the same set of delay defects, pattern $P1$ would screen more defects than $P2$ as it will affects more longer paths.

In our pattern selection process, each pattern is investigated to determine the number of active endpoints. An endpoint which observes a transition is referred to as an *active endpoint*. An endpoint which does not observe a transition, referred to as *non-active*, has no contribution to the fault coverage. Figure 5 shows the breakup of endpoints for six patterns, generated using multiple-detect technique, into three different categories: 1) non-active endpoints, 2)

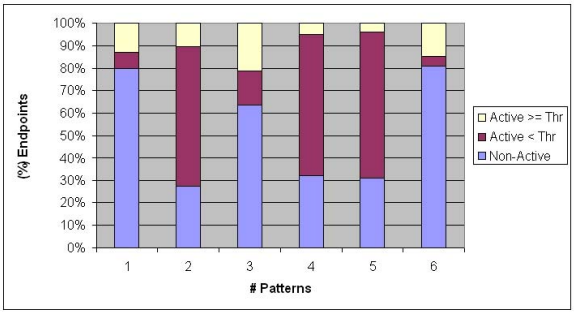

Figure 5: Endpoint analysis for six patterns generated using multiple-detect for s38584 benchmark.

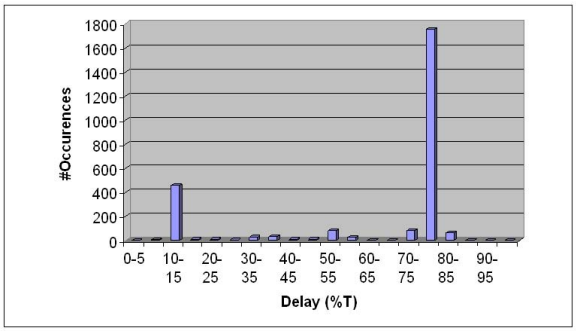

Figure 6: Path delay distribution after pattern selection from a 15-detect pattern set (benchmark s38584).

active endpoints with path length less than a threshold limit and 3) active endpoints with affected paths length greater than the threshold limit. The threshold limit is defined as the maximum slack of the path length region which also implies the minimum small delay defect size that can be detected. For our experiment, we fixed the threshold limit as the LP-cutoff limit as 30% cycle time and SP-cutoff limit as 70% cycle time. Patterns 2, 4 and 5 have higher percentage of active endpoints. This criteria alone does not ensure that all active endpoints are observing long path delays. Consider patterns 1, 3 and 6 respectively which have less number of active endpoints. For these patterns, the percentage of active endpoints observing long paths is very high, although the overall percentage of active endpoints is low. Such patterns ensure that most of the faults are detected through long paths. The path delay distribution of the resulting pattern set using the pattern selection process is shown in Figure 6. The variation of path delays is much smaller and concentrated in the long path range, although a small percentage of short paths are activated.

5. EXPERIMENTAL RESULTS

The entire automation flow consisting of various steps is shown in Figure 7. The complete process can be divided into two phases: 1) Pre-processing phase and 2) Pattern generation and selection.

5.1 Pre-processing Phase

Before starting the test pattern generation process, the design is pre-processed for path delay analysis. This is performed using a static timing analysis tool (Synopsys PrimeTime[7]) and the endpoints are classified into different path length categories based on the least slack reported for each endpoint (*Step 1*). Table 2 shows the total number of endpoints (column 2) and the endpoints with their least slack path for five ISCAS'89 benchmark circuits in each of the different regions. In our experiment, we used the cutoff limit

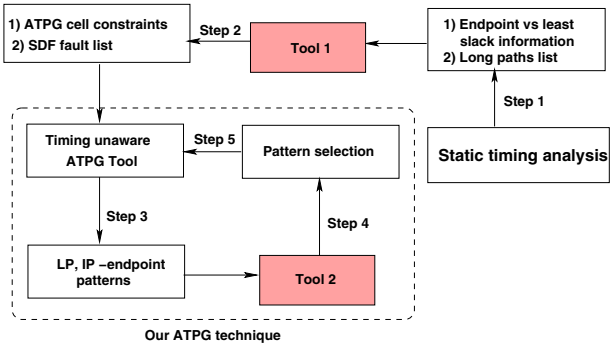

Figure 7: Automation Flow.

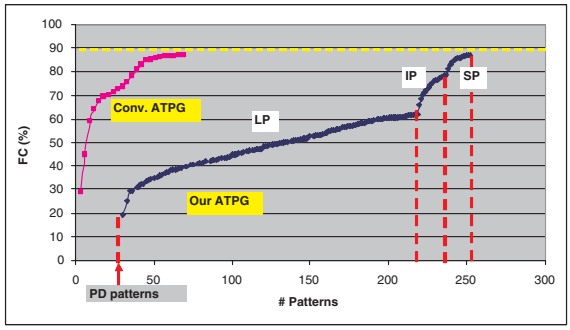

Figure 8: Conventional and our pattern generation process for SDF fault list (s38584 benchmark).

Table 2: Number of endpoints with least slack in each path delay region.

Design	Total	LP	IP	SP	NP	GDFs	SDFs
s13207	626	10	138	453	25	15084	160
s15850	516	46	202	246	22	16178	230
s35932	1728	32	545	1149	2	44366	320
s38417	1564	208	735	600	21	44986	2120
s38584	1276	157	770	342	7	52874	2124

Table 3: ATPG results.

Design	*SDFs*	*GDFC %*	*Conv. ATPG*		*Our ATPG*		
			Patt	*Time [sec]*	*SDFC %*	*Patt*	*Time [sec]*
s13207	160	92.5	27	29	90.62	33	116
s15850	230	73.04	20	42	51.74	27	131
s35932	320	100	16	73	83.12	29	155
s38417	2120	99.43	63	114	98.87	174	534
s38584	2124	86.86	69	140	79.75	252	1768

for LP-region (LP-cutoff) as slack less than 30% of cycle period. Similarly, the SP-cutoff was set to slack greater than 70% of cycle period. It can be noticed that majority of the endpoints for circuits s13207, s15850 and s35932 have their least slack path in the SP region. While for circuits s38417 and s38584, it is in the IP region. The NP-endpoints are endpoints with a path starting from the primary input. Since, the primary inputs are held constant during the delay fault test generation due to low-cost tester speed limitations, *NP* endpoints do not contribute to delay fault coverage.

Since our focus is to detect all small delay defects on long paths, we also extract all the paths in LP-region using static timing analysis (*Step 1*). These paths are converted to a fault list which will be used during the pattern generation process. This is performed by identifying all fault locations along each long path and a slow-to-rise and slow-to-fall fault is considered for each fault site. In case of multiple clock domains, the following path analysis and fault set extraction needs to be performed for each individual clock domain. Table 2 also shows the total number of gross delay faults (GDFs) on all paths, i.e. total transition faults and the small delay faults (SDFs) which lie only along the long paths. For example, for benchmark s38584, the static timing analysis tool gave 1491 long paths with a LP-cutoff of 30% cycle time. We, then extracted all the fault sites along the 1491 long paths to obtain 2124 transition faults, which forms our small delay fault list (see Table 2, last column).

We have developed a Perl program (*Tool 1*) which takes as input the worst slack information for each endpoint, along with design, clocking and cutoff limits for LP and SP region, and generates the atpg cell constraints for each path region (LP, IP and SP) in *Step 2*. As explained earlier, these cell constraints are required to ignore the value captured in the respective endpoints. However, an endpoint with a cell constraint can be loaded with a valid bit to activate and propagate a fault site. These cell constraints are used during the test pattern generation process. The tool also performs the extraction of transition faults (SDFs) using long paths list.

5.2 Pattern Generation and Selection

In the pattern generation process, we perform our ATPG technique (*Step 3, 4 and 5*) for LP- and IP-endpoints only, as observe points one at a time, respectively. We used a commercial no-timing ATPG tool (Synoposys TetraMax [7]). Initially we perform path delay pattern generation for the long paths. The generated path delay pattern set covers a subset of faults in the SDF list which lie along the detectable paths. For example, for ISCAS'89 benchmark s38584, the path delay test pattern set for long paths provided 19.28% coverage of SDF transition fault list (2124 faults).

In the next step (*Step 3*), for the remaining faults in the SDF fault list after fault grading path delay patterns, a 15-detect pattern generation with only LP-endpoints observable, followed by IP-endpoints only being observable is performed. The generated LP (IP) pattern sets are analyzed (*Step 4*), to measure the path delay distribution for the respective endpoints in each pattern set. The patterns are then re-ordered based on the percentage of active endpoints affected, as explained in Section 4. and a subset of the patterns with very high percentage of active endpoints is selected. Also, we mask the endpoints affected by short paths in each pattern, i.e. a delay less than 30% of the cycle time. This ensures that no short paths are exercised in the pattern set. We have developed another software program (*Tool2*), which performs the path delay distribution analysis, followed by masking of endpoints exercising short paths. It also re-orders the pattern set using the ATPG tool (TetraMAX).

Since, the short paths are masked in the pattern set, the resulting pattern set is re-fault simulated (*Step 5*) to get the accurate coverage of SDF fault list. Figure 8 compares the fault coverage graphs of our method and the conventional timing unaware ATPG. As shown, we initially generate the path delay patterns (PD-patterns) followed by LP-patterns, and IP-patterns based on LP- and IP-endpoints being observable, respectively. Note that, multiple-detect technique during LP-endpoints pattern generation may not cover all possible long paths. This may require a higher multiple-detect pattern generation and it will increase the run time. Hence, if it is possible to increase the frequency without any adverse affects of test power and IR-drop, we generate IP-patterns for the remaining faults in the SDF fault list and higher frequency can be applied to detect them. After each step of fault detection, the detected faults are removed from the fault list. For example, after fault grading path

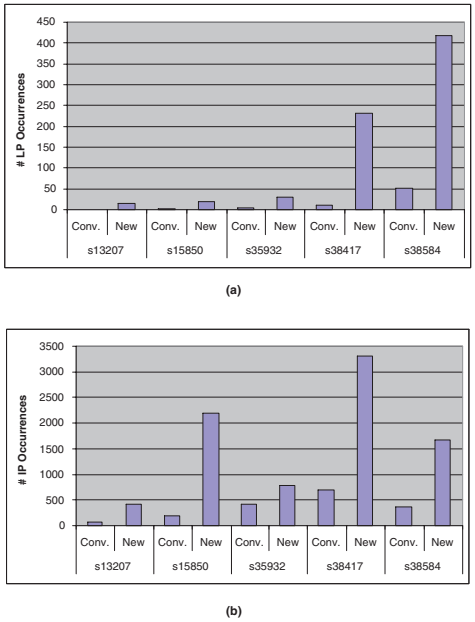

Figure 9: Comparison of number of paths affected by conventional and our technique, (a) long and (b) intermediate paths, for the same small delay fault coverage.

delay patterns for transition fault model, we removed the detected faults from the faut list. Finally, after removing the detected faults by LP- and IP-patterns, we perform conventional ATPG using SP-endpoints. These faults in the SDF list covered by pattern generation using SP-endpoints are actually gross delay faults. The conventional no-timing ATPG generates less number of patterns because all the endpoints are observable and it can easily activate and propagate the fault through short paths. Therefore, the delay defect size required for such a pattern set will be very high.

Table 3 shows the ATPG results for the SDF fault list using conventional no-timing ATPG and our technique. **Note that, the fault coverage numbers obtained from the conventional ATPG does not reflect the defect size coverage of each fault in contrast to our method.** It is simply the gross delay defect size coverage (GDFC, shown in column 3), reported for the SDF fault list. As shown in Figure 8, after LP- and IP-patterns we performed pattern generation for SP-endpoints to get the final gross delay coverage of our technique. The small delay fault coverage (SDFC), obtained by our technique, is shown in column 6 using LP- and IP-patterns only. This coverage is excluding the additional coverage achieved using SP-patterns. The number of patterns are higher for our method due to limited number of endpoints being observable (to increase the probability of affected long paths) in our technique. The best way to compare our method with the traditional timing unaware ATPG is to observe the long and intermediate paths exercised. Figures 9 (a) and (b) shows the comparison of long and intermediate paths respectively for all benchmarks for the two methods. It can be noticed that the pattern set generated by the new technique affects higher number of long and intermediate paths in all benchmark circuits. This shows the effectiveness of the proposed technique in affecting more longer paths for small delay faults. In future work, we will evaluate the effectiveness of our method on real silicon.

6. CONCLUSION

In this paper, we have proposed new test pattern generation and pattern selection techniques to target small delay faults on long paths in deep-submicron designs. The technique divides the scan flip-flops into different categories based on the least slack. It then performs ATPG on each category to exercise more longer paths and is very efficient in detecting small delay defects. The experimental results showed that the proposed technique can detect a significant number of small delays through longer paths which otherwise would have escaped using gross delay pattern set. As a result, this will increase the reliability of the designs.

7. ACKNOWLEDGEMENT

This work was supported in part by Semiconductor Research Corporation under contract SRC 2005-TJ-1322.

8. REFERENCES

[1] 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.

[2] V. Jayaram, J. Saxena and K. Butler, Scan-Based Transition-Fault Test Can Do Job, *EE Times*, Oct. 2003.

[3] 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.

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

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

[6] Cadence Inc., "http://www.cadence.com,", 2005.

[7] Synopsys Inc., "User Manuals for SYNOPSYS Toolset Version 2004.06," Synopsys, Inc., 2004.

[8] H. Hao and E.J. McCluskey, "Very-low-voltage testing for weak CMOS logic ICs," in Proc. *Int. Test Con. (ITC'93)*, pp. 275-284, 1993.

[9] R. Foster, "Why Consider Screening, Burn-In, and 100-Percent Testing for Commercial Devices?," *IEEE Transactions on Manufacturing Technology*, vol. 5, no. 3, pp. 52-58, 1976.

[10] P. Gupta and M. S. Hsiao, "ALAPTF: A new transition fault model and the ATPG algorithm," in Proc. *Int. Test Conf. (ITC'04)*, pp. 1053-1060, 2004.

[11] A. K. Majhi, V. D. Agrawal, J. Jacob, L. M. Patnaik, "Line coverage of path delay faults," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 5, pp. 610-614, 2000.

[12] W. Qiu, J. Wang, D. M. H. Walker, D. Reddy, X. Lu, Z. Li, W. Shi and H. Balichandran, "K Longest Paths Per Gate (KLPG) Test Generation for Scan-Based Sequential Circuits," in Proc. *Int. Test Conf. (ITC'04)*, pp. 223-231, 2004.

[13] B. Kruseman, A. K. Majhi, G. Gronthoud and S. Eichenberger, "On hazard-free patterns for fine-delay fault testing," in Proc. *Int. Test Conf. (ITC'04)*, pp. 213-222, 2004.

[14] J. Saxena, K. M. Butler, V. B. Jayaram, N. V. Arvind, P. Sreeprakash and M. Hachingerr, "A Case Study of IR-Drop in Structured At-Speed Testing," in Proc. *Int. Test Conf. (ITC'03)*, pp. 1098-1104, 2003.

[15] J. Rearick and R. Rodgers, "Calibrating Clock Stretch During AC Scan Testing," in Proc. *Int. Test Conf. (ITC'05)*, 2005.

[16] B. Benware, C. Schuermyer, N. Tamarapalli, Kun-Han Tsai, S. Ranganathan, R. Madge, J. Rajski and P. Krishnamurthy, "Impact of multiple-detect test patterns on product quality," in Proc. *Int. Test Conf. (ITC'03)*, pp. 1031-1040, 2003.

[17] B.N. Lee, L. C. Wang and M. S. Abadir, "Reducing pattern delay variations for screening frequency dependent defects," in Proc. *VLSI Test Symp. (VTS'05)*, pp. 153-160, 2005.