2010 19th IEEE Asian Test Symposium

Circuit Topology-Based Test Pattern Generation for Small-Delay Defects*

Sandeep Kumar Goel[1†] Krishnendu Chakrabarty[1] Mahmut Yilmaz[2] Ke Peng[3] Mohammad Tehranipoor[3]

[1]Dept. Electrical & Computer Engg.
Duke University
Durham, N.C. 27708, USA
{skgoel, krish}@ee.duke.edu

[2]Advanced Micro Devices, Inc.
1 AMD Place
Sunnyvale, Calif. 94085, USA
mahmut.yilmaz@amd.com

[3]Dept. Electrical & Computer Engg.
University of Connecticut
Storrs, Conn. 06269, USA
{kpeng, tehrani}@engr.uconn.edu

Abstract

For sub-nanometer designs, testing for small-delay defects (SDDs) is essential to achieve low defect escapes for the manufactured silicon. Existing solutions for testing SDDs are not practical for high-volume production environments due to large pattern count or long compute time, or both. In this paper, we present a production-friendly method that takes the circuit topology into account while generating patterns for SDDs. Experimental results on several IWLS'05 benchmark and six industrial circuits show that compared to the default timing-aware pattern set; the proposed method reduces pattern count an average of 172% for IWLS benchmarks and an average of 105% for industrial circuits. We demonstrate the production-worthiness of our approach by using several quality metrics and showing that the proposed method provides similar or higher coverage for SDDs compared to the default timing-aware ATPG, but only with a significantly small number of test patterns and in significantly small run time.

1. Introduction

Advances in design methods and process technology are continuing to push the envelope for the integrated circuits. The use of advanced process technology brings forward several design and test challenges. In addition to manufacturing defects such as resistive opens/bridges, design-related issues such as process variation, power supply noise, crosstalk, and design-for-manufacturability (DfM) rule violations such as butted contacts, insufficient via enclosures introduce small additional delays in the circuit [1][2]. These delays are commonly referred to as small-delay defects (SDDs), and testing them is one of the major challenges that the semiconductor industry is facing today. SDDs can cause immediate failure of a circuit if introduced on critical paths, whereas they cause major quality concerns if they occur on non-critical paths. For very high-quality products (0-100 DPPM), testing SDDs is a must.

Conventional test methods such as stuck-at test and transition-delay fault test [3] still provide very high coverage of most manufacturing defects, but cannot guarantee sufficient coverage of SDDs. Stuck-at-based tests do not target delay defects, while the traditional transition fault tests target gross-delay defects and they are likely to miss SDDs. Path-delay-based test can provide good coverage of SDDs; however, generation of path-delay patterns suffers from practical problems such as enumeration of long paths and test generation complexity with increasing numbers of paths.

Testing SDDs requires fault activation and propagation through long paths in the design. Several techniques [2][5-16] have been proposed in literature for testing SDDs and they can be classified into three basic categories: (1) faster-than-at-speed, (2) timing-aware ATPG, and (3) pattern/fault selection based hybrid techniques. In faster-than-at-speed techniques, existing or modified transition-delay fault patterns are applied at a clock speed higher than the system speed. Applying patterns at faster clock speed than the rated system speed reduces the available slack in the design and helps detect SDDs. Although faster-than-at-speed methods provide very good coverage for SDDs, they suffer from several drawbacks such as higher power consumption during test, complex test application, and chances of hazards that can cause unexplainable failures.

Timing-aware ATPG techniques for testing SDDs are preferred among EDA companies [5][9][10]. In timing-aware ATPG, delay information for every node is taken into account while generating test patterns. In these techniques, for each fault, the ATPG tool tries to excite and propagate fault effects along the longest testable path possible. Timing-aware patterns do not require test application at higher-than-system speed, although if applied can also detect some reliability defects. Some of the major drawbacks of timing-aware ATPG are huge pattern count and long compute time [6][16]. Due to these problems, timing-aware ATPG is not very suitable for large industrial designs.

Effective pattern selection based on metrics such as output deviation has been proposed in [6]. Here instead of generating timing-aware patterns, effective patterns are selected from a pool of patterns. Unfortunately, the quality of the results obtained from this method depends on the quality of the pool of patterns, which is hard to determine prior to pattern selection. Recently, it was shown in [16] that traditional transition fault patterns already provide good SDD coverage for a large number of faults in the circuit, and for only a fraction of the total number of nodes, should timing-aware ATPG be carried out. However, the two techniques presented in [16] also fall short in terms of achieving low pattern count for large industrial designs. Therefore, efficient methods are required that provide high coverage of SDDs with minimum increase in pattern count while maintaining maximum coverage for gross-delay defects. This is also required for cases when the entire timing-aware pattern set cannot be loaded on the tester due to tester memory limitation, and test pattern truncation needs to be carried out.

In this paper, we present circuit topology-based test pattern generation for SDDs. Similar to the two approaches in [16], the proposed approach identifies a subset of faults for which timing-aware ATPG should be carried out, while, for rest of the node traditional transition fault patterns can be generated. The identification of a subset of faults is done based on the fan-out count of

*The work of S.K. Goel and K. Chakrabarty was supported in part by SRC under Contract no. 1588 and by NSF under Grant no. ECCS-082385. The work of K. Peng and M. Tehranipoor was supported in part by SRC under Contracts 1455 and 1587, and by NSF under Grant no. ECCS-0823992.

†S.K. Goel is currently with TSMC, NA.

1081-7735/10 $26.00 © 2010 IEEE
DOI 10.1109/ATS.2010.59

nodes in the design. The main motivation behind this selection is that a SDD on a high fan-out node is more critical as its fault effect can be propagated through several paths and cause circuit failure. For several IWLS benchmarks [18], we show that the proposed method leads to only a small increase in the pattern count and provide high coverage of gross- and small-delay defects. As only a fraction of faults are considered for timing-aware ATPG, the proposed approach also results in run-time savings.

The remainder of this paper is organized as follows. Section 2 provides the motivation behind the proposed approach and the fault selection method. Section 3 presents the circuit topology-based test pattern generation flow. Experimental results for several IWLS'05 benchmark circuits that demonstrate the effectiveness of our proposed approach are presented in Section 4. Section 5 shows the applicability of the proposed approach for six industrial circuits. Section 6 concludes the paper.

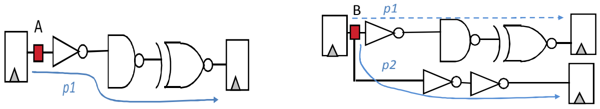
2. Circuit Topology-Based Fault Selection

Although timing-aware ATPG is the preferred solution for SDD testing due to easy test pattern generation and application flow, it has not been adopted widely due to large pattern count and long compute time. In the timing-aware mode, most ATPG tools try to excite and propagate a delay-fault effect along the longest path (also the least slack path) for the corresponding fault. Due to the inherent problem of enumerating a large number of paths for each fault, timing-aware ATPG results in significantly higher run time.

Recently, it has been shown [16] that not all faults should be considered for timing-aware ATPG; as there is no change in the path length for a majority of faults (up to 80%) even if timing-aware ATPG is carried out for them. For these faults, the ATPG tool is able to propagate a fault effect through the longest path fortuitously or because there are limited number of paths. That is also the main motivation behind the work presented in this paper.

The challenge here is to identify a minimum subset of faults in a circuit that requires timing-aware ATPG. The fault set should not be too large because that will lead to large pattern count and run time. Also, it should not be too small; otherwise the improvement in SDD quality will be small and difficult to measure. In [16], a subset of faults was selected based on the difference in tested path length and longest path possible for each fault. The tested and longest path lengths for each fault were obtained by performing timing-aware simulation of traditional transition fault patterns. Therefore, the fault selection was heavily dependent on the results reported by the ATPG tool. In our approach, we look at the faults from circuit topology point of view and no ATPG is required to select the fault list.

In a given circuit, nodes with multiple fan-outs are potential targets for timing-aware ATPG. This is due to the fact that if a defect occurs at any of these nodes, the fault effect can be propagated through multiple paths (related to fan-outs) and can affect the circuit functionality. However, the non-timing-aware ATPG may select the easiest path for these nodes and SDDs may go undetected.



(a) (b)

Figure 1: Examples of nodes with different fan-outs.

To understand this, consider the two example nodes shown in Figure 1. For node A, there is only one propagation path possible because it does not have any fan-out. Therefore, if ATPG is able to generate a test for it, it will be through path p1, which is also the longest and only path for this node. Therefore, whether we do timing-aware ATPG or non-timing-aware ATPG, coverage of SDD for this node will not change. Now if we look at node B (shown in Figure 1(b)), there are two paths possible. Path p1 is longer than path p2. For these types of nodes, performing timing-aware ATPG can improve the SDD coverage. Furthermore, if these nodes have been already tested for SDDs, other nodes in the circuit will also have higher coverage for SDDs if they use paths through these nodes for fault excitation/propagation.

To further improve the coverage of SDDs, instead of selecting one long path for each fault, multiple long paths can be selected. This can be done easily by including all fan-out nodes in the list of timing-aware target faults. Consider the example shown in Figure 2. Node A has three fan-outs and therefore we select node A, A1, A2, and A3 when we consider node A. Similarly, for node C, both C1 and C2 nodes are also considered as timing-aware target faults. If all fan-out branches are tested through their longest paths, the fan-out stem will be tested through multiple long paths.

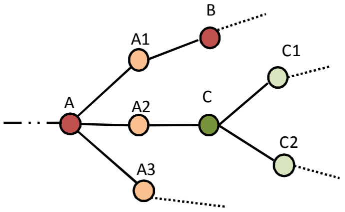


Figure 2: Fan-out node selection for multiple-path coverage.

Figure 3 shows the fan-out distribution of nodes for various IWLS'05 benchmark circuits [18]. As expected, the number of nodes decreases with increase in the fan-out count. There are very few nodes with a fan-out count of six or higher. Therefore, to avoid very small fault subset when selecting faults as timing-aware candidates, the fan-out count should be limited to four or five only and not higher.

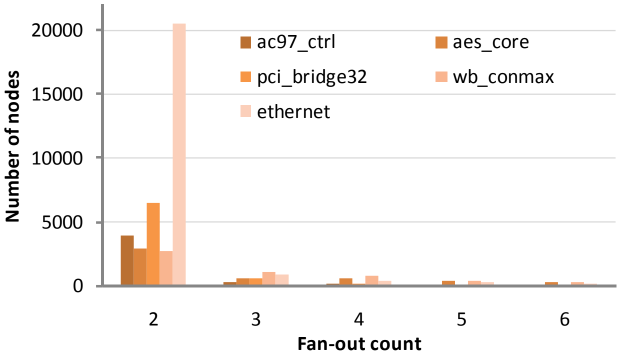


Figure 3: Nodes' fan-out profile for IWLS benchmarks.

The extraction of fan-out count for nodes in a circuit is very easy and does not require any functional or timing simulation. Therefore, the run time impact of calculating fan-outs can be ignored. Next we describe the ATPG flow that uses the fan-out-based fault selection criterion as presented in this section. Because the fan-out information is used in the proposed method for deciding which faults should be targeted for timing-aware ATPG, we also refer to this methodology as the *fan-out-aware method.*

3. SDD Pattern Generation

The fan-out-aware SDD APTG flow is shown in Figure 4. The proposed flow is a hybrid one and it requires both timing-aware and non-timing-aware transition fault ATPG. First, fan-out counts for all nodes in the design are determined. Next, based on a user-defined limit *fc*, nodes with fan-out count equal to or higher than *fc* are selected for timing-aware ATPG. For each selected node, all the corresponding fan-out nodes are also selected (as shown in Figure 2). Timing-aware ATPG is then performed on the selected nodes and fan-out-aware SDD patterns are generated. Next, these patterns are fault simulated for the non-timing-aware target faults that have been filtered earlier. Note that this is a non-timing-aware fault simulation. For the remaining undetected faults, traditional transition fault patterns are generated. The combined pattern set for this methodology includes the fan-out-aware SDD patterns and the top-off-TDF patterns.

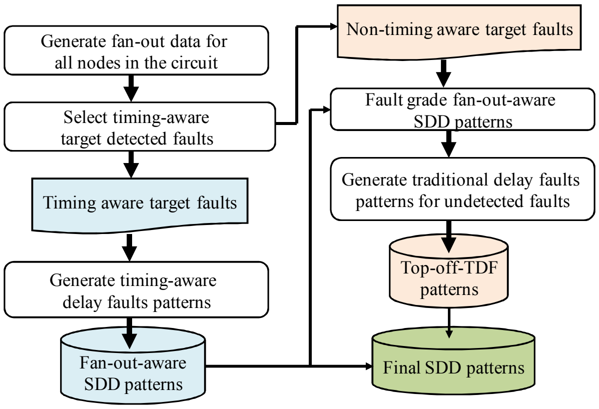

Generate fan-out data for all nodes in the circuit
Select timing-aware target detected faults
Timing aware target faults
Generate timing-aware delay faults patterns
Fan-out-aware SDD patterns
Non-timing aware target faults
Fault grade fan-out-aware SDD patterns
Generate traditional delay faults patterns for undetected faults
Top-off-TDF patterns
Final SDD patterns

Figure 4: Fan-out-aware SDD pattern generation flow.

For further reduction in test pattern count, several variations of the proposed flow are possible. For example, TDF patterns can be generated first and fault simulated in timing-aware mode for timing-aware target faults. Next, only for undetected timing-aware target faults, timing-aware patterns can be generated, followed by non-timing-aware pattern generation for the remaining undetected faults. However, a complete discussion on all possible variations of the proposed flow is not considered here.

4. Experimental Results

In this section, we present detailed experimental results for several IWLS benchmarks. We compare the test quality and pattern effectiveness based on several different metrics: (1) delay test coverage (DTC) [5], (2) total number of long paths excited by the pattern set, (3) length of the longest path excited by the pattern set, and (4) number of unique SDD locations covered by the pattern set. We also discuss the DPPM impact of the proposed method in terms of detecting randomly injected SDDs. In our experiments, we have used a commercial ATPG tool; however, any other ATPG tool that supports timing-aware pattern generation can be used. Also note that, for timing-aware pattern generation, we have used fault dropping based on the slack margin concept [5] with slack margin value of 5%. However, similar results were also obtained for other slack margin limits.

Table 1 shows the basic test characteristics for the selected IWLS benchmarks. All these circuits were physically synthesized and place-and-routed to obtain realistic delay values. Columns 4, 5, and 6 show the number of test patterns, transition fault coverage (TC), and delay test coverage (DTC) for the non-timing-aware transition fault patterns (TDF) (launch-on-capture), respectively.

Circuit	Gate	Total	TDF		
	count	Faults	Patterns	TC (%)	DTC (%)
wb_dma	7619	68084	173	87.30	80.31
tv80	13326	59172	823	97.15	77.46
systemcaes	17817	94056	357	91.80	84.94
mem_ctrl	22015	98998	564	94.92	85.95
usb_funct	25531	155904	288	95.97	87.74
ac97_ctrl	28083	175070	249	95.45	90.26
aes_core	29186	165368	670	95.50	87.16
dma	41026	165556	656	95.48	87.32
pci_bridge32	43907	309216	437	93.61	86.08
wb_conmax	59484	347300	593	91.11	84.94
ethernet	153948	868248	2508	97.31	90.42

Table 1: Basic test characteristics of IWLS benchmarks.

4.1 Delay Test Coverage

First, we show the variation in the test pattern count and DTC for the proposed method with the number of fan-outs (*fc*). For four large IWLS circuits, Figure 5 and 6 show the overall pattern count and DTC obtained from the proposed method with *fc* = 2, 3, and 4, respectively. From the figures, we can see that both overall test pattern count as well as DTC decreases with an increase in fan-out count. This is to be expected as the number of timing-aware target faults decreases with an increase in fan-out count (Figure 4), resulting in fewer timing-aware patterns and lower DTC.

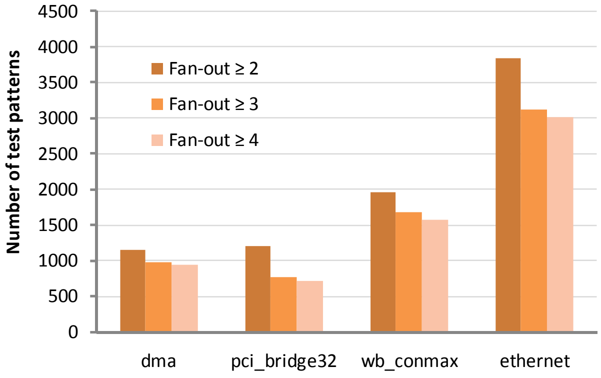


Figure 5: Pattern count variation with fan-out count.

Next, we compare the ATPG results obtained from the proposed method to the default timing-aware ATPG on the entire circuit (TA-SDD), and the two fault-filtering-based approaches presented in [16]. As the approaches presented in [16] outperform the timing-critical-path-based fault filtering approach [15], comparison with timing-critical-path-based fault filtering is not necessary. Table 2 shows the test pattern and DTC results for these approaches. Columns 2 and 3 show the pattern count and DTC for TA-SDD respectively. Columns 4, 5, 6, and 7 show the same but for the two approaches: TDF + Top-off –SDD (Approach 1) and Top-off-SDD +Top-off-TDF (Approach 2) as described in [16].

Columns 8 and 9 show the pattern count and DTC for the proposed method with fan-out count two or more.

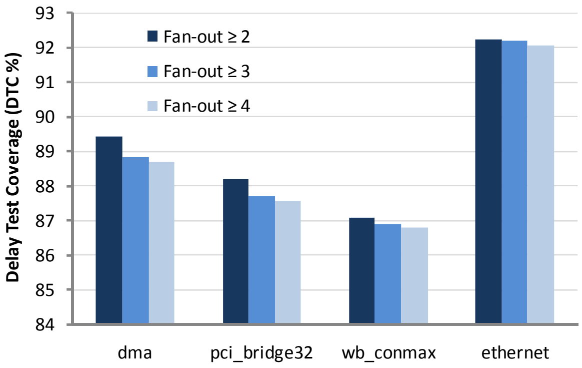


Figure 6: Variation in DTC with fan-out count.

From the table, we can see that, for almost all circuits, the proposed method results in the lowest pattern count. The DTC values obtained from the proposed methods are also similar (except circuit tv80) to those obtained from other methods. The bottom row (Average) in the table shows weighted average (based on gate count) percentage increase in pattern count and weighted average absolute improvement in DTC obtained from respective approaches compared to the traditional non-timing-aware TDF (Table 1). Note that we take the weighted average in order to accurately reflect the benefits that are obtained for larger circuits.

Circuit	TA-SDD		[16]				Our Method	
			Approach 1		Approach 2		Fan-out ≥ 2	
	Pat	DTC	Pat	DTC	Pat	DTC	Pat	DTC
wb_dma	319	82.12	409	82.11	286	81.78	254	81.61
tv80	1746	82.17	2444	83.03	1751	82.34	1307	80.74
systemcaes	1192	87.18	1166	87.15	870	86.74	720	86.42
mem_ctrl	1108	89.07	1512	89.34	1116	88.76	832	88.25
usb_funct	981	90.79	616	90.39	487	90.04	653	90.1
ac97_ctrl	825	91.87	481	91.31	382	91.07	501	91.37
aes_core	1646	90.14	1565	90.25	1048	89.03	1131	89.33
dma	1625	90.16	1518	90.29	1023	89.02	1144	89.42
pci_bridge32	2147	88.84	1390	88.62	1132	88.3	1199	88.21
wb_conmax	2739	87.77	1995	87.67	1598	87.23	1964	87.08
ethernet	8726	93.56	9342	93.44	7188	93.14	3841	92.25
Average	**245%**	**2.9**	**246%**	**2.8**	**165%**	**2.3**	**73%**	**2.0**

Table 2: Test pattern and DTC results.

From the average results, we can see that default timing-aware ATPG (TA-SDD) results in 245% average increase in pattern count compared to TDF patterns, while the average absolute improvement in DTC is 2.9. Approach 1, as presented in [16], performs worse than TA-SDD pattern with 246% increase in pattern count and only 2.8 increase in DTC. Approach 2 results in 165% increase in pattern count with an average increase of 2.3 in DTC. The proposed method outperforms other methods and results in only 73% increase in pattern count compared to TDF, with an average increase of 2 in DTC. Compared to TA-SDD, it shows a saving of 172% in pattern count while the loss in DTC is only 0.9.

As mentioned earlier, one of the major hurdles in industry-wide adaptation of timing-aware patterns is the huge pattern count. Industry is looking for solutions that give maximum increment in delay coverage with minimum increase in pattern count without any loss of transition test coverage (gross-delay defects). This is also required for cases when the entire timing-aware pattern set cannot be loaded on the tester due to tester memory limitation, and test pattern truncation needs to be carried out. To measure the effectiveness of a method in such cases, a DTC efficiency metric was presented in [16]. Delay test coverage efficiency (DTCE) is defined as the average improvement in DTC per 1% increase in overall pattern count compared to the traditional TDF pattern set. This is an important metric because what we really want is a method that can improve coverage beyond what TDF patterns have already achieved. Therefore, higher DTCE value is desired for an effective method. Figure 7 shows DTCE comparison for the methods shown in Table 2 as well as for the fan-out-aware method with fan-out count 3 and 4. From Figure 8, we can see that the proposed fan-out-aware method outperforms TA-SDD as well as previously published approaches. The DTCE value for the fan-out-aware method increases with an increase in fan-out count.

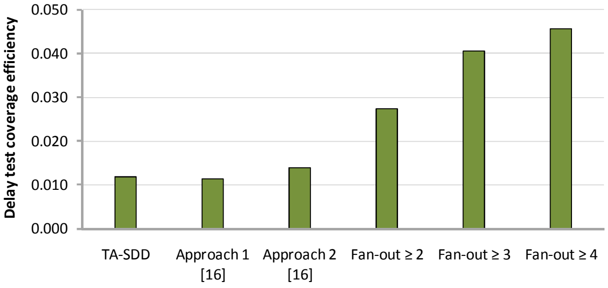


Figure 7: Delay test coverage efficiency comparison.

4.2 Number of Unique Long Paths

The second metric we use to show the effectiveness of our approach is the number of unique long paths excited by the pattern set. This is an important metric because the excitation of more number of long paths implies possible detection of more SDDs. In the rest of the experiments, we only compare our approach with the default timing-aware pattern set (TA-SDD), which provides the maximum DTC value as shown in the previous section. For the five large IWLS circuits, Figure 8 shows the total number of unique paths with lengths greater than or equal to 90% of the clock period that are excited by the TA-SDD pattern set and the proposed method with different fan-out values.

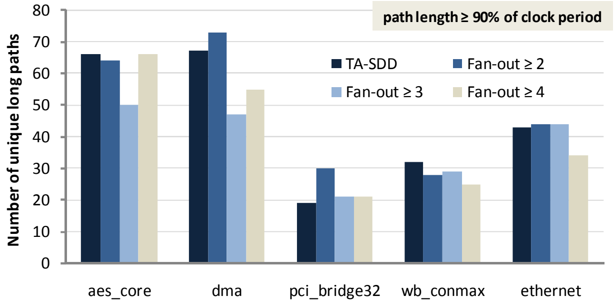


Figure 8: Comparison of the number of unique long paths between TA-SDD and fan-out-aware methods.

It can be noted from Figure 10 that the number of unique long paths excited by the fan-out-aware method decreases with an increase in fan-out count. This is to be expected because the number of test patterns decreases with increase in fan-out count, resulting in fewer excited paths. We can also see that the fan-out-aware method typically excites more long paths than the full timing-aware pattern set (TA-SDD). This is especially good considering the difference in the total number of patterns between TA-SDD and the fan-out-aware method. This also means the lower DTC value for the proposed method, as shown in Table 2, may be due to less coverage of short paths instead of long paths.

4.3 Length of Longest Path

The third metric we use to show the effectiveness of our approach is the length of the longest path excited by the pattern set. This is also an important metric because the excitation of the longest path is directly related to the size of the SDD that can be detected in a circuit given the system speed. Figure 9 shows the comparison of the longest path length excited by the TA-SDD pattern set and the proposed fan-out-aware method. Again we can see that the proposed method typically excites longer or similar paths. Considering the difference in the overall pattern count (as shown in Table 2), this is quite significant.

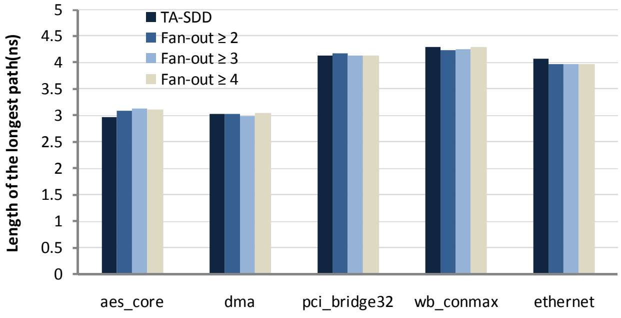


Figure 9: Longest-path length comparison for TA-SDD and the proposed method.

4.4 Number of Unique SDDs

Next we compare the number of unique SDD locations (gate pins) that are on the long paths and also covered by the pattern set. Figure 10 shows the number of unique SDDs (on paths with lengths greater than or equal to 90% of the clock period) for the two methods. From the figure, we can see that fan-out-aware method results in a similar or higher number of unique SDDs.

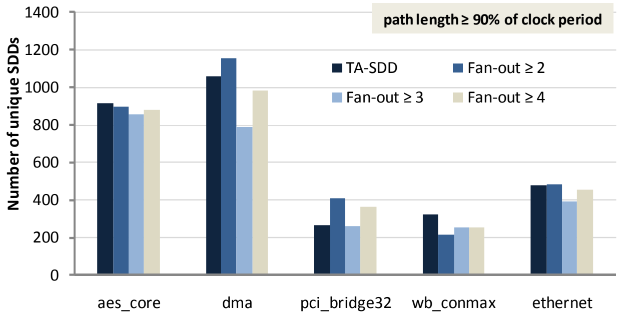


Figure 10: Comparison of number of unique SDDs between TA-SDD patterns and the proposed method.

Based on the comparison using four different metrics, it can be concluded that the proposed fan-out-aware approach delivers significant reduction in pattern count and provides better or similar test quality for SDDs compared to timing-aware pattern set. Another major advantage of the proposed method is very low compute time. Because only a few nodes are considered for time-consuming timing-aware pattern generation, the proposed method requires significantly less compute time (60% to 80%) than TA-SDD patterns set as shown in Figure 11.

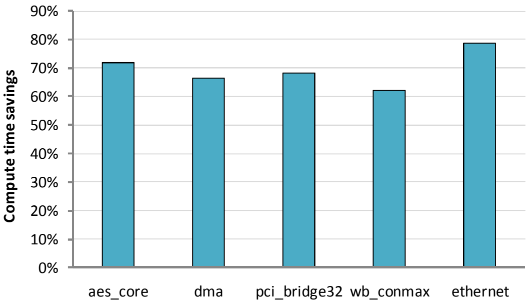


Figure 11: Compute time savings from the proposed method.

4.5 Random Fault Injection and Detection

To show the added value of the SDD patterns over the traditional TDF patterns, as well as to show quality effectiveness of the proposed approach and TA-SDD patterns, we use random fault injection and verification. In our experiments, we inject a large number (50,000) of random SDDs of different sizes (size less than or equal to 10% of clock period) on randomly selected nodes in each circuit. Note that we only insert one defect at a time in a circuit. Next, fault simulation of TDF, full timing-aware ATPG patterns (TA-SDD), and the proposed fan-out-aware patterns for different fan-out values are carried out on each defective circuit.

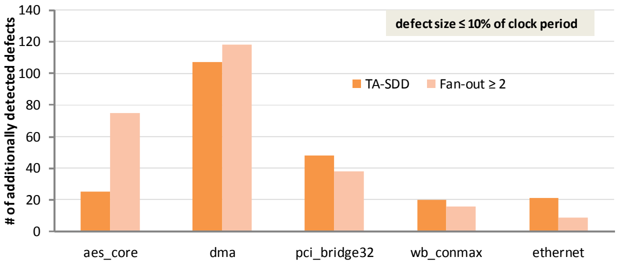
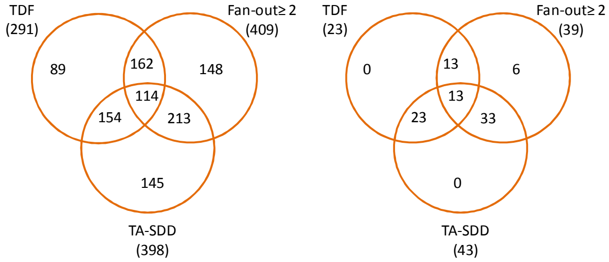


Figure 12: Number of additionally detected SDDs.

Compared to TDF patterns, the number of additionally detected defects for TA-SDD and fan-out-aware patterns are shown in Figure 12. We can clearly see the fan-out-aware method outperforms the TA-SDD pattern set. The quality impact of fan-out-aware method is 180 to 2,360 DPPM, while for the full timing-aware SDD pattern set it is 420 to 2,160 DPPM.

To further understand the added quality value of SDD patterns, overlap between the different defects detected by TDF, TA-SDD, and fan-out-aware method needs to be studied. Figure 13 shows the Venn diagrams for detected defects for the three methods. From Figures 13, we can see that defect overlap between TDF and fan-out-aware patterns is larger than the overlap between TDF and SDD patterns. This is to be expected, since the fan-out-aware pattern set contains a large number of TDF patterns and only a small number of timing-aware patterns. However, the number of defects uniquely detected only by the fan-out-aware method is similar to the number detected by full timing-aware patterns.



(a) dma (b) wb_conmax

Figure 13: Detected defects overlap.

As TA-SDD and fan-out-aware methods result in a large number of unique defects, effective patterns can be selected from the two pattern sets to create the pattern set that has the highest quality.

5. Industrial Circuits

In this section, we demonstrate the production worthiness of our approach by showing the savings obtained by the proposed method compared to TA-SDD pattern set for several industrial circuits. Table 3 shows percentage increase in pattern count and absolute improvement in DTC for TA-SDD and fan-out-aware method compared to the TDF pattern set. The weighted averages are listed in the last low. From the table, we can see that fan-out-aware method results in 68% (263−195 = 68) to 105% (163−158 = 105) savings in pattern count compared to the TA-SDD patterns set. The impact on DTC is less than 0.5% for fan-out count greater than or equal to 2, and less than 1% for fan-out count greater than or equal to three. The proposed method also results in 10% to 46% reduction in compute time compared to TA-SDD.

Ckt.	Gate	TA-SDD		Fanout ≥ 2		Fanout ≥ 3	
	Count	PAT	DTC	PAT	DTC	PAT	DTC
Ckt. A	45132	157%	4.90%	138%	4.35%	120%	3.85%
Ckt. B	264086	241%	3.05%	164%	2.60%	145%	2.45%
Ckt. C	291246	257%	3.06%	206%	2.70%	150%	2.29%
Ckt. D	293589	255%	3.09%	192%	2.69%	147%	2.28%
Ckt. E	321920	234%	5.56%	186%	4.95%	148%	4.49%
Ckt. F	525984	308%	3.48%	215%	2.95%	185%	2.72%
Average		**263%**	**3.70%**	**195%**	**3.22%**	**158%**	**2.89%**

Table 3: Results comparison for industrial circuits.

6. Conclusion

Testing of small-delay defects (SDDs) is becoming a serious concern today and the problem is likely to be aggravated for newer technologies. Existing solutions for testing SDDs are not practical enough for high-volume production environments due to large pattern count, or large compute time, or both. In this paper, we have proposed a production-friendly method that takes the circuit topology into account while generating patterns for SDDs. The proposed method selects a small number of nodes in the circuit for which the timing-aware pattern generation needs to be carried out, while for other nodes in the circuit, conventional transition fault patterns can be generated.

Experimental results on several IWLS'05 benchmark and six industrial circuits show that compared to the default timing-aware pattern set, the proposed method reduces pattern count an average of 172% for IWLS benchmarks and an average of 105% for industrial circuits. The average penalty in absolute delay test coverage (DTC) is less than 1% for both IWLS and industrial circuits. However, by using other metrics such as number of long paths, length of the longest path, and the number of unique SDDs, we have shown that the proposed method outperforms the default timing-aware ATPG, and the loss in DTC is attributed to low coverage of short paths that are only important from a reliability point of view. Furthermore, random fault injection and verification of 50,000 SDDs shows that the proposed method achieves similar or higher quality for SDDs as does default timing-aware ATPG and requires only a small number of test patterns. This is very important from the point of view of high-volume production.

7. References

[1] S. Mitra et al., Delay Defect Screening Using Process Monitor Structures, In Proc. VLSI Test Symposium, 2004.

[2] B. Kruseman et al., On Hazard-free Patterns for Fine-Delay Fault Testing, In Proc. International Test Conference, 2004.

[3] J.A. Waicukauski et al., Transition Fault Simulation, IEEE Design and Test of Computers, 1987.

[4] I. Pomeranz and S.M. Reddy, On N-detection Test Sets and Variable N-detection Test Sets for Transition Faults, In Proc. International Test Conference, 1999.

[5] X. Lin et al., Timing-Aware ATPG for High Quality At-Speed Testing of Small Delay Defects, In Proc. Asian Test Symposium, 2006.

[6] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, Test-Pattern Grading and Pattern Selection for Small-Delay Defects, In Proc. VLSI Test Symposium, 2008.

[7] R. Putman and R. Gawde, Enhanced Timing-Based Transition Delay Testing for Small Delay Defects, In Proc. VLSI Test Symposium, 2006.

[8] Y. Sato et al., Invisible Delay Quality – SDQM Model Lights Up What Could Not Be Seen, In Proc. International Test Conference, 2005.

[9] A. Uzzaman et al., Not all Delay Tests Are the Same – SDQL Model Shows True-Time, In Proc. Asian Test Symposium, 2006.

[10] R. Kapur, J. Zejda, and T.W. Williams, Fundamentals of timing information for test: How simple can we get?, In Proc. International Test Conference, 2007

[11] H. Li, Z. Li, and Y. Min, Reduction in number of paths to be tested in delay testing, Journal of Electronic Testing: Theory and Application (JETTA), Vol. 16, Number 5, 2000.

[12] N. Ahmed, M. Tehranipoor, and V. Jayram, Timing-based delay test for screening small delay defects, In Proc. Design Automation Conference, 2006.

[13] R. Turakhia, et al., Silicon Evaluation of Longest Path Avoidance Testing for Small Delay Defects, In Proc. International Test Conference, 2007.

[14] H. Yan and A.D. Singh, Evaluating the Effectiveness of Detecting Delay Defects in the Slack Interval: A Simulation Study, In Proc. International Test Conference, 2004.

[15] X. Lin, M Kassab, and J. Rajski, Test Generation for Timing-Critical Transition Faults, In Proc. Asian Test Symposium, 2007.

[16] S.K. Goel, N. Devta-Prasanna, and R. Turakhia, Effective and Efficient Test Pattern Generation for Small Delay Defects, In Proc. VLSI Test Symposium, 2009.

[17] N. Devta-Prasanna et al., Accurate Measurement of Small Delay Defect Coverage of Test Patterns, In Proc. International Test Conference, 2009.

[18] IWLS 2005 Benchmarks, http://iwls.org/iwls2005/benchmarks.html