A Framework of High-quality Transition Fault ATPG for Scan Circuits

Seiji Kajihara, Shohei Morishima,
Akane Takuma, Xiaoqing Wen
Kyushu Institute of Technology,
Iizuka, Japan
{kajihara, morishima, akane, wen}@cse.kyutech.ac.jp

Toshiyuki Maeda, Shuji Hamada,
Yasuo Sato
Semiconductor Technology Academic Research Center,
Yokohama, Japan
{maeda, hamada.shuji, satoh.y}@starc.or.jp

ABSTRACT
This paper presents a framework of high-quality test generation for transition faults in full scan circuits. This work assumes a restricted broad-side testing as a test application method for two-pattern tests where control of primary inputs and observation of primary outputs are restricted. Because we use a modified time expansion model of a circuit-under-test during ATPG and fault simulation, conventional ATPG and fault simulation programs can work with minor change. The proposed ATPG method consists of two algorithms, which are activation-first and propagation-first, and for each fault it is decided which algorithm should be applied. Test patterns are generated such that transition faults with small delay can be detected, i.e. a path for fault excitation and propagation becomes as long as possible. In experimental results we evaluate test patterns generated by the proposed method using SDQM that calculates delay test quality, and show the effectiveness of the proposed method.

1. Introduction
Delay testing is getting more and more important for deep sub-micron circuits as the increase of defects that affect timing behavior of logic circuits [1], [2]. For example, resistive opens or resistive bridges increase delay of signal lines. It is also known that crosstalk changes signal propagation delay. And power supply noise and process variation, which may not be classified as defects, affect the circuit delay [3],[4].

When test patterns are generated for delay testing, either path delay fault model [5] or transition fault model [6] is typically considered. Since a path delay fault targets testing for a critical path, the ability of detection of small delay on the path is high. However, it is difficult to determine critical paths. And it is practically impossible to test all paths in a circuit because the number of paths in the circuit is sometimes huge.

On the other hand, transition faults can cover the circuit comprehensively, and ATPG (Automatic Test Pattern Generation) for transition faults can be performed easily by extending ATPG for stuck-at faults. One of the disadvantages of transition fault testing is that delay size caused by a defect is not considered. Since the detectable delay size depends on generated test patterns, the detection of small delay is not guaranteed in ATPG.

In recent delay testing, detection of error due to the small delay is an important issue. Higher test quality is achieved by detecting the small delay. Test generation for transition faults with the small delay has been considered in [7],[8]. In [7], a PODEM-based ATPG algorithm is employed, and a test pattern is generated under predetermined path sensitization conditions for fault activation and fault propagation, respectively. Although the conditions correspond to either single path sensitization or multiple path sensitization, it is difficult to find the longest sensitizable path under multiple path sensitization. In [8], SAT-based ATPG is proposed, but it considers the longest sensitizable path under single path sensitization condition. Therefore if multiple path sensitization is needed to detect faults, this method would not find the longest paths.

In this paper we present a framework of high-quality test generation for transition faults in a full scan circuit. When applying a two-pattern test to a conventional scan circuit, the second pattern is not allowed to shift-in. This work assumes a broad-side (launch-off-capture) testing [9] with restriction for the observation of primary outputs and the control of primary inputs. During ATPG and fault simulation we use a modified time expansion model of a circuit-under-test for the restricted broad-side testing such that a conventional ATPG program can work with minor change. The ATPG method, proposed in this paper, generates test patterns for transition faults with small delay, i.e., paths for fault excitation and propagation are sensitized as long as possible. The proposed ATPG method consists of two different ATPG algorithms, which are activation-first and propagation-first. For each fault it is decided which algorithm should be applied. In experimental results we evaluate test patterns generated by the proposed method using SDQM [10, 11] that calculates delay test quality, and show the effectiveness of the proposed method.

1-4244-0292-1/06/$20.00

This paper is organized as follows. In Section 2, we explain transition delay testing and Statistical Delay Quality Model (SDQM) to evaluate test quality of test patterns. In Section 3, we give a circuit model which we treat in ATPG and fault simulation. In Section 4, we explain the proposed ATPG algorithms. Section 5 shows experimental results for benchmark circuits, and Section 6 concludes this paper.

2. Preliminaries

2.1 Application of two-pattern tests

In order to detect a delay fault, a two-pattern test that consists of successive two test vectors is required. In case of a transition fault, the first pattern initializes the circuit to allow signal transitions at the fault site, and the second pattern detects an error caused by the fault at an observable output. For example, to detect a falling transition fault at line *l*, the first pattern is required to set logic value 1 at line *l* and the second pattern is required so as to detect a stuck-at 1 fault on *l*.

Since detection of delay faults requires more constraints of logic values in test patterns than stuck-at faults, it is difficult to achieve high fault coverage by random patterns. Hence ATPG and scan design are essential even for delay testing. When applying a two-pattern to a conventional scan circuit, the second pattern is not allowed to shift-in because the second pattern must be captured at-speed. There are two well-known methods for delay testing, which are broad-side testing (launch-off-capture) [9] and skewed-load testing (launch-off-shift) [12]. The broad-side testing uses two capture clocks for the circuit at-speed. The skewed-load testing applies one capture clock just after a scan clock. Since it is difficult to apply different clock signals with an interval corresponding at-speed, the broad-side testing is widely used. This work assumes the broad-side testing as a test application method for two-pattern tests.

2.2 SDQM

The statistical delay quality model (SDQM) [10, 11] has been proposed for the evaluation of delay test quality. The SDQM is generated by first assuming a delay defect distribution that is based on the actual defect probability in a fabrication process, and then investigating the sensitized transition paths and calculating their delay lengths. Detectable delay defect sizes are defined as the difference between the test timing and the path lengths. Finally, the probability of detecting small delay defects is calculated by multiplying the distribution probability for each defect. The calculated value is called the statistical delay quality level (SDQL).

We explain the SDQM using Fig. 1 and Fig. 2. Assume that the delay of the longest testable path that detects a transition fault is 5 *ns* and the delay of the sensitized path through which the fault is detected by generated test patterns is 4 *ns*. And the system clock timing T_{MC} is 6 *ns* and the test timing T_{TC} is 7 *ns*. For the fault, detectable delay size T_{det} is 3 (= 7-4) *ns*. Define the difference between the delay size of longest testable path and T_{MC} as T_{mgn}. If delay size of a fault is less than T_{mgn}, the fault is untestable. In this case, T_{mgn} is 1 (=6-5) *ns*. And if delay size is greater than 1 and less than 3, the fault remains undetected.

Depending on the delay size for transition faults, fault coverage is changed, i.e. the percentages of undetectable faults, detectable faults and undetected faults are changed. Fig. 2 shows an example graph of fault coverage. If the area indicating undetected is small, it means test quality of test patterns is high. The SDQL corresponds to the area indicating undetected when the distribution probability of delay size is uniform.

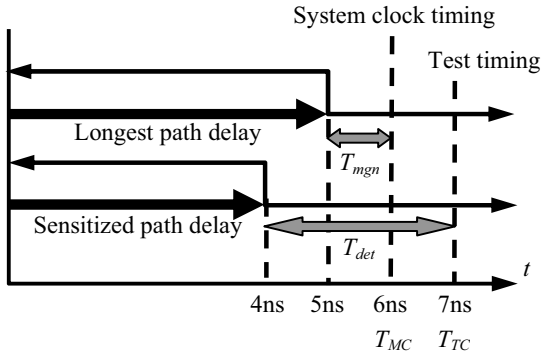

Fig. 1. Detectable delay size

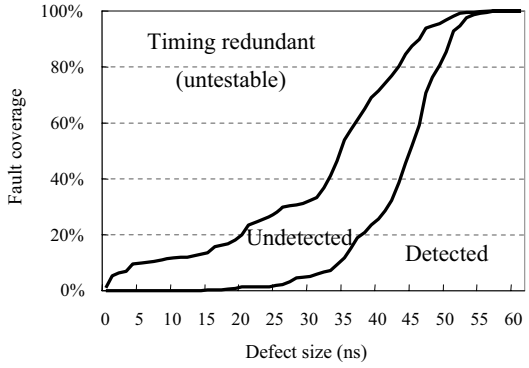

Fig. 2. Example of SDQL graph

3. Netlist Transformation for ATPG

3.1 Controllable Inputs and Observable Outputs during At-Speed Testing

In ATPG for a scan circuit, an output of a scan flip-flop is regarded as a pseudo primary input (PPI), and an input of a

scan flip-flop is regarded as a pseudo primary output (PPO). As we assume the broad-side testing in this work, PPIs of the first patterns are controllable and PPOs of the second patterns are observable. Though primary inputs may be always controllable, it is difficult to change values at the primary inputs between two capture clocks. Hence we assume that the primary input values cannot be changed between the first pattern and the second pattern, that is, the second pattern take the same primary input values as the first pattern. In addition, as it is difficult to observe values at the primary outputs during at-speed testing, we assume that primary outputs are not observable. In the rest of this paper, we refer to these conditions as a *restricted broad-side* condition.

3.2 Time Expansion Circuit for the Restricted Broad-side Testing

For broad-side testing, we can employ a combinational ATPG by using a time expansion model of the circuit-under test. Under the restricted broad-side condition, a fault is not treated as detected even if the fault effect is propagated to a primary output. We remove lines and gates which never reach to any pseudo primary output from the circuit model for test generation. In addition, we connect each primary input between the first time frame and second time frame so that primary input values are not changed in a two-pattern test. Thus, a time expansion circuit is transformed as illustrated in Fig. 3. By using such a modified netlist, we need not to change programs of ATPG and fault simulation for the restricted broad-side testing.

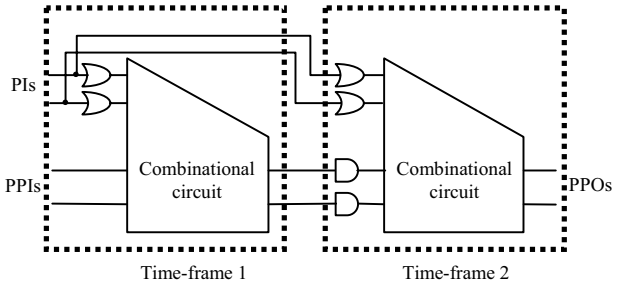

Fig. 3. Time expansion circuit for restricted broad-side testing

4. ATPG algorithms

4.1 Algorithm selection

In order to detect a transition fault with small delay, the fault should be activated and propagated through long paths. Fault activation paths from at least one PPI to the fault site have to propagate a signal transition at the PPI, while fault propagation paths from the fault site to a PPO don't have to propagate the signal transition at the fault site. Thus the conditions of path sensitization are different between fault activation and fault propagation. When we consider the longest path for a fault, total path length should be maximized. In addition, multiple path sensitization should be considered for both fault activation and fault propagation.

In the proposed ATPG method, we prepare two algorithms: a propagation-first algorithm and an activation-first algorithm. These algorithms, whose details are described in the next section, are based on SOCRATES [13]. The propagation-first algorithm finds a fault propagation path as long as possible. After fixing the fault propagation path, the fault is activated. The activation-first algorithm finds a fault activation path as long as possible. After fixing the fault activation path, find a fault propagation path by applying a part of the propagation-first algorithm.

In ATPG, which algorithm is applied is decided for each fault. While the propagation-first algorithm selects a long propagation path among many paths from the fault site to PPOs, the effort of finding the long path would be meaningless if the difference of length of propagation paths is same. But the propagation-first algorithm is useful when the difference of propagation path length is large. Similarly, we don't have to care the length of activation paths in case that any activation path has similar length to other activation paths. For fault f, we decide the ATPG algorithm to be applied according to the following procedure:

(1) Calculate the length of the structurally longest propagation path for f, $p\_max(f)$, and the length of the structurally shortest propagation path for f, $p\_min(f)$.

(2) Calculate the length of the structurally longest activation path for f, $a\_max(f)$, and the length of the structurally shortest activation path for f, $a\_min(f)$.

(3) If $p\_max(f) - p\_min(f) > a\_max(f) - a\_min(f)$, then apply the propagation-first ATPG for f. Otherwise apply the activation-first ATPG.

Fig. 4 shows an example. Suppose a fault with $p\_max(f) = 8$, $p\_min(f) = 6$, $a\_max(f) = 7$, and $a\_min(f) = 3$. Because of $p\_max(f) - p\_min(f) \leq a\_max(f) - a\_min(f)$, the activation-first ATPG is applied for f.

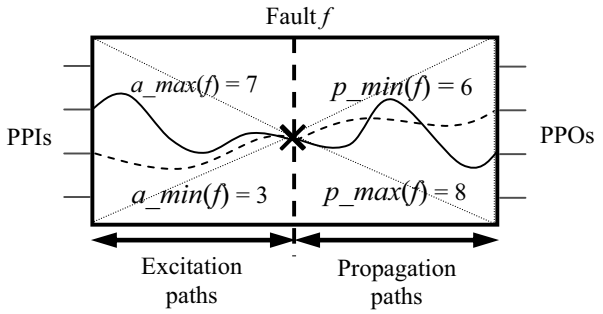

Fig. 4. Algorithm selection

4.2 Propagated-first algorithm

The propagation-first ATPG algorithm, which is based on SOCRATES [13], first sensitizes a fault propagation path from the fault site to a PPO, and then justifies the value for fault activation. In ATPG, the fault propagation path is sensitized by extending a D-frontier until one of D-frontiers is reached to a PPO. In general there are many D-frontiers and an ATPG algorithm usually chooses one which is the closest to a PPO or one which is easy to observe according to an observability measure such as SCOAP [14]. Such a criterion for D-frontier selection does not result in a long fault propagation path. When a D-frontier is selected, the propagation-first ATPG algorithm chooses one such that the long fault propagation path would be sensitized.

Fig. 5 explains the criteria of the propagation-first algorithm to select a D-frontier in the following example in Fig. 5. For fault *f*, let *dis*(*i*) be the path length from the fault site to D-frontier *i*, and let *p_max*(*i*) be the length of the structurally longest propagation path from *i* to PPOs. Note that path length is calculated as the number of gates on the path. D-frontier *i* is selected so that value of D-frontier *dis*(*i*) + *p_max*(*i*) is maximum. Such a D-frontier can produce longer fault propagation paths than other D-frontiers potentially. In the case of Fig. 6, D-frontier *a* is selected among D-frontiers {*a, b, c*}.

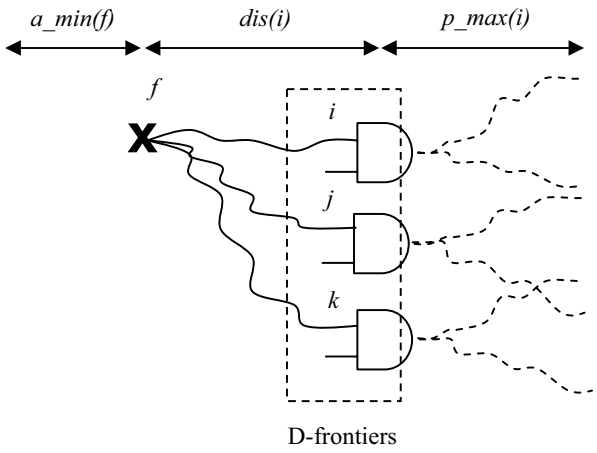

Fig. 5. Distance of D-frontiers

D-frontier	*dis(i)*	*p_max(i)*
a	6	9
b	4	6
c	7	7

Fig. 6. Example of selecting a D-frontier

4.3 Activation-first algorithm

The activation-first algorithm finds a fault activation path as long as possible. After fixing the fault activation path, it sensitizes a fault propagation path with the same manner as the propagation-first algorithm. The activation-first algorithm has the following feature:

- The length of the fault activation path to be found is designated first.
- The branch and bound algorithm based on the depth-first search from the fault site is performed.
- When a path is extended, the implication procedure of ATPG is performed to assign logic values. When a conflict occurs in the implication procedure, do backtrack and search another path.
- Every time a path is extended, it is confirmed that there remains at least one sensitizable fault propagation path from the fault site to a PPO. This is similar to X-path check of ATPG.

4.4 Procedure

We give the procedure of the proposed ATPG below.

(1) Read a netlist of a circuit-under-test.

(2) Transform the netlist to a netlist of the modified time expansion circuit corresponding to the restricted broad-side testing.

(3) Create a transition fault list *F*.

(4) For each line *f* in *F,*

(5) calculate *p_max*(*f*), *p_min*(*f*), *a_max*(*f*), *a_min*(*f*).

(6) Initialize test pattern set $T = \varnothing$.

(7) For delay size *DS* = 0 to *ML*,

(8) For each fault *f* in *F*

(9) if *f* with *DS* has not been detected yet,

(10) if *p_max*(*f*)-*p_min*(*f*) > *a_max*(*f*) -*a_min*(*f*),

(11) then *t* = propagation_first_ATPG(*f*).

(12) else apply *t* = activation_first_ATPG(*f*).

(13) if $t \neq \varnothing$, then fault_simulation(*t*) and add *t* to *T*

(14) Calculate_SDQL(*T*).

Note that *ML* at (7) is a predetermined maximum delay size.

5. Experimental results

We implemented the proposed test generation method on UltraSPARCIII 1.5GHz, 1024MB using C language, and applied for ISCAS'89 benchmark circuits. In calculating SDQL, we assume that delay of every gate is 0.2 *ns*.

Table 1 shows results of test generation for transition faults. In order to show the effectiveness of the proposed method, we run a conventional ATPG algorithm that was based on SOCRATES, and compare the results with the proposed ones. Note that the conventional ATPG was also developed in-house, and the proposed ATPG was obtained by modifying the conventional one. In Table 1, the second column shows the number of transition faults. The 3rd to 5th columns give results of the conventional transition ATPG and the 6th to 8th columns give results of the proposed ATPG. The columns "Tests" show the number of generated two-pattern tests. The proposed method generated more test patterns than the conventional method. This is because the proposed method deals with a fault as "detected" when it is detected through a long path. Therefore the chance of accidental detection of faults in fault simulation is reduced in the proposed method. The columns "Coverage" and "Efficiency" show fault coverage and fault efficiency calculated by normal transition fault simulation, i.e., the defect size assumed in fault simulation is large enough for detecting the faults. Fault efficiency is defined as a percentage of detected faults in faults not identified as untestable. The number of detected faults and the number of untestable faults identified by the ATPG algorithms are given in the columns "Detect" and "Untest", respectively. Note that these numbers are slightly different between two methods because the search order in ATPG is different. Fault coverage and fault efficiency of both methods were almost same. Due to the restricted broad-side testing, faults which have no fault propagation path to PPOs or no fault activation path from PPIs are untestable. Therefore fault coverage could not be high. However more than 99% fault efficiency were obtained.

Table 2 shows how the proposed method improves SDQL. It means that the lower the SDQL value is, the higher the delay test quality is. The second and third columns show SDQL values for test patterns generated by the conventional method and the proposed method, respectively. The SDQL values can be improved by calculating the length of the longest sensitizable path for each fault more accurately. Although we used the structural (topological) longest paths as the longest sensitizable path, the structural longest paths are unsensitizable in many cases. We tried an additional experiment in which the longest path for each fault is calculated after setting necessary assignments to detect the fault. Since we can avoid selecting some unsensitizable paths as the longest path, we can obtain better SDQL values. Note that calculating more accurate longest path length does not affect test patters to be generated. The results are shown in column "Proposed+" of Table 2. From these results, it is observed that the proposed ATPG can contribute to improvement of test quality, but it is important for SDQM to measure the longest path length accurately.

The graph in Fig. 7 shows fault coverage for each delay defect size for s1423. The upper one of four curves goes down by calculating the length of the longest sensitizable path. The lowest curve goes up by generating test patterns with higher test quality. Because either improvement contributes to reduction of "undetected" area, SDQL values could be decreased.

6. Conclusions

In this paper we proposed a framework of high-quality test generation for transition faults in full scan circuits. During ATPG and fault simulation we used a modified time expansion circuit for the restricted broad-side testing such that conventional ATPG and fault simulation programs can work with minor change. The proposed ATPG method consisted of two algorithms, which are activation-first and propagation-first, and for each fault it is decided which algorithm should be applied. In experimental results we evaluated test patterns generated by the proposed method using SDQM, and showed the effectiveness of the proposed method for high test quality.

Table 1. Results of ATPG

Circuits	Faults	Conventional method					Proposed method				
		Tests	Detect	Untest	Coverage	Efficiency	Tests	Detect	Untest	Coverage	Efficiency
s1423	2244	162	1926	316	85.78%	99.90%	222	1925	316	85.82%	99.84%
s5378	4948	412	4198	750	84.84%	100.00%	477	4198	750	84.84%	100.00%
s9234	10618	1004	8632	1961	81.30%	99.71%	1130	8632	1961	81.30%	99.71%
s13207	14796	986	11780	3012	79.61%	99.97%	1046	11779	3011	79.62%	99.95%
s15850	17568	775	12343	5221	70.26%	99.97%	976	12344	5219	70.26%	99.96%
s35932	53340	149	44012	9328	82.51%	100.00%	230	44012	9328	82.10%	100.00%
s38417	48988	2378	48029	959	98.01%	100.00%	4222	48013	959	98.04%	99.97%
s38584	52112	2441	43701	8401	83.86%	99.98%	3021	43701	8377	83.86%	99.92%

Table 2. SDQL of generated test patterns

Circuits	SDQL (ppm)			Ratio	
	Conventional	Proposed	Proposed+	Proposed	Proposed+
s1423	0.317	0.296	0.237	93.4%	74.8%
s5378	0.331	0.300	0.297	90.6%	89.7%
s9234	1.370	0.991	0.938	72.3%	68.5%
s13207	1.532	1.490	0.370	97.3%	24.2%
s15850	1.470	1.416	1.032	96.3%	70.2%
s35932	23.983	23.536	17.558	98.1%	73.2%
s38417	0.511	0.413	0.162	80.8%	31.7%
s38584	0.877	0.845	0.419	96.4%	47.8%
Average				90.7%	60.0%

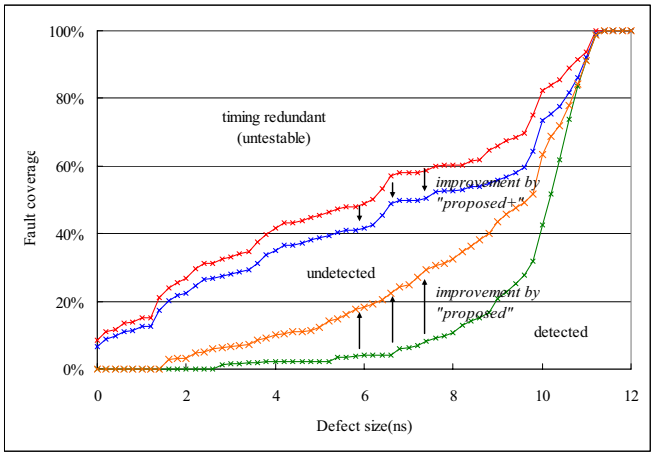

Fig. 7. Effects of the proposed method for s1423

ACKNOWLEDGMENTS

This work was supported in part by the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] B. Kruseman, A. K. Majhi, G. Gronthoud, and S. Eichenberger, “On hazard-free patterns for fine-delay fault testing,” *International Test Conference*, pp. 213–222, 2004.

[2] S. Mitra, E. Volkerink, E. McCluskey, and S. Eichenberger, “Delay defect screening using process monitor structures,” *VLSI Test Symposium*, pp. 43–52, 2004.

[3] L.-C. Chen, S. K. Gupta and M. A. Breuer, “High Quality Robust Tests for Path Delay Faults”, in Proc. *VLSI Test Symp.*, pp.88-93, April 1997.

[4] K.-T. Cheng, S. Dey, M. Rodgers, K. Roy. “Test Challenges for Deep Sub-Micron Technologies,” Design Automation Conf., pp.142-149, June 2000.

[5] G. L. Smith, "Model for delay faults based upon paths," *Int'l Test Conf.*, pp.342-349, 1985..

[6] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, V. S. Iyengar, "Transition fault simulation," *IEEE Design and Test of Computers*, pp. 32-38, April 1987.

[7] Y. Shao, I. Pomeranz, S. Reddy, "On Ggenerating High Quality Tests for Transition Faults," *Proc. Asian Test Symposium*, pp. 1-8, 2002.

[8] K. Yang, K.-T Cheng, L.-C Wang, "TranGen: A SAT-Based ATPG for Path-Oriented Transition Faults," *Proc. Asian and South Pacific Design Automation Conference*, pp. 92-97, 2004.

[9] J.Savir, "On broad-side delay testing," *Proc. VLSI Test Symposium*, pp.284-290, 1994.

[10] Y. Sato, S. Hamada, T. Maeda, A. Takatori, and S. Kajihara, “Evaluation of the statistical delay quality model,” *Asian and South Pacific Design Automation Conference*, pp. 305–310, 2005.

[11] Y. Sato, S. Hamada, T. Maeda, A. Takatori, Y. Nozuyama, S. Kajihara, "Invisible delay quality – SDQM model lights up what could not be seen,” *International Test Conference*, 47.1, Nov. 2005.

[12] J. Savir, "Skewed-Load Transition Test: Part I, Calculus," *International Test Conference*, pp. 705-713, 1992.

[13] M. Schulz et al., “SOCRATES: A Highly Efficient Automatic Test Generation System,” *IEEE Trans. on CAD.*, pp. 126-137, Jan. 1988.

[14] L. H. Goldstein, E. L. Thigen, "SCOAP: Sandia Controllability/Observability Analysis Program," *Design Automation Conference*, pp. 190-196, 1980.