

Signature Rollback – A Technique for Testing Robust Circuits

Uranmandakh Amgalan¹, Christian Hachmann¹, Sybille Hellebrand¹, Hans-Joachim Wunderlich²

¹University of Paderborn, Germany

²University of Stuttgart, Germany

Abstract

Dealing with static and dynamic parameter variations has become a major challenge for design and test. To avoid unnecessary yield loss and to ensure reliable system operation a robust design has become mandatory. However, standard structural test procedures still address classical fault models and cannot deal with the non-deterministic behavior caused by parameter variations and other reasons. Chips may be rejected, even if the test reveals only non-critical failures that could be compensated during system operation. This paper introduces a scheme for embedded test, which can distinguish critical permanent and non-critical transient failures for circuits with time redundancy. To minimize both yield loss and the overall test time, the scheme relies on partitioning the test into shorter sessions. If a faulty signature is observed at the end of a session, a rollback is triggered, and this particular session is repeated. An analytical model for the expected overall test time provides guidelines to determine the optimal parameters of the scheme.

1 Introduction

In nanoscale CMOS parameter variations have become a major challenge [4, 15]. Static variations caused by processing and mask imperfections as well as dynamic variations, including for example power density and temperature variations or variations in power supply voltage caused by IR drop, may cause delay variations [13]. Additional timing problems arise from glitch propagation or clock shifts at the boundaries between multiple clock domains. Furthermore, device degradations and an increased susceptibility of system operation to external disturbances (“soft errors”) cause increasing reliability problems [3, 4, 14]. A robust design style has thus become mandatory to improve yield (“design for yield”, DFY) and manu-

facturability (“design for manufacturability”, DFM) as well as to reach acceptable reliability and availability levels (“design for reliability”, DFR). The goal in robust circuit design is to find a good trade-off between desired design features, quality and cost rather than working with worst case assumptions.

Recent approaches try to effectively combine affordable solutions at all levels of the design. A prominent example is Razor register, where flip-flops are combined with shadow registers to deal with delay-errors [6, 7]. Delay-error detection is accomplished by comparing flip-flops with the contents of the shadow latches, and the backup values in the shadow latches can be used for rollback and recovery after a timing error. This time redundancy is exploited to allow an aggressive scaling of the supply voltage, where error rates up to several thousand errors per second are considered to support a low power system operation. Another application of time redundancy is the GRAAL architecture featuring a level sensitive design with two non-overlapping clocks [15]. Here delay-error detection is possible even without extra latches, and rollback and recovery is supported by extra backup flip-flops. Other approaches address soft error mitigation, as for example the schemes proposed in [12, 14].

While these advanced design techniques ensure a robust behavior, testing still has to address the circuit structure to identify permanent defects. However, standard structural test procedures cannot deal with the non-deterministic behavior caused by parameter variations or soft errors. Chips may be rejected during manufacturing test, even if the test reveals only failures that could be compensated during system operation. This problem of “overtesting” has already been addressed in the context of error tolerance and in the context of functional delay testing [5, 10, 16, 18-20]. Here automatic test pattern generation is tuned, such that only patterns for critical or realistic failures, respectively, are generated. In contrast to that, the scheme presented in this paper addresses robust designs based on time redundancy. It works with standard test sets and distinguishes whether a failure indication is due to a critical permanent fault or to a non-critical temporary problem.

This work has been performed within the framework of the DFG-project RealTest (DFG-grants HE 1686/3-1 and WU 345/5-1).

A straightforward technique to distinguish permanent from temporary failures is to repeat tests with faulty outcomes. In case of a permanent failure the same faulty result can be observed again, whereas for a temporary failure the second test is likely to produce a different result. However, as the proposed scheme mainly targets future scenarios with failure rates, which may be much higher than the typical rates observed in the past, simply repeating the complete test has a two-fold disadvantage. Obviously, the probability of a temporary failure increases with the test time. For high failure rates this implies that even several iterations of the test may fail due to temporary failures. As a consequence, either a large number of repetitions become necessary, or yield loss due to “type II errors” rejecting acceptable devices still remains a problem. To overcome these problems, the scheme presented in this paper partitions the test into several sessions. If a faulty signature is observed at the end of a session, a rollback is triggered, and only this particular session is repeated. An analytical model for the expected overall test time provides guidelines to adjust the parameters of the scheme, such that the best trade-off between hardware overhead, yield improvement and test time is obtained.

2 Signature rollback

Structural testing can still effectively characterize circuits with time redundancy, if it is possible to distinguish between permanent and temporary failures. To achieve this goal, the proposed technique partitions the test into sessions and triggers a rollback, if a session results in a failure indication. In the sequel, the implementation of this idea is described in more detail for an embedded test based on the STUMPS architecture [1]. As sketched in Figure 1, the method does not depend on a specific test pattern generator.

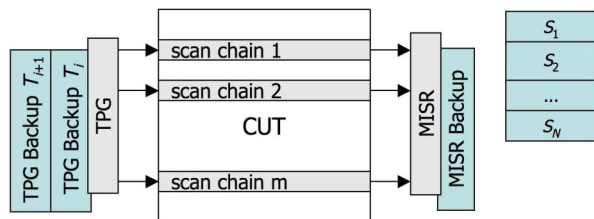


Figure 1: Architecture for signature rollback.

The TPG-block can be an LFSR combined with a phase shifter or a more advanced pattern generator embedding or decompressing deterministic patterns, e.g. described in [2, 8, 9, 17].

To realize the test with rollback, a given test T with X patterns is partitioned into N sessions T_1, \dots, T_N , and without loss of generality it is assumed that all sessions

have $x = \lceil X/N \rceil$ patterns. For each session T_i the correct signature S_i must be determined and made available during test, either by storing it on chip or by loading it from the ATE.

During test, the i -th session starts with storing the initial state of the MISR in the backup register. Then the patterns for session T_i are generated and the test responses are compacted with the MISR. When the last response is shifted out, the first pattern of the next session is already shifted in. Therefore the state of the test pattern generator must be saved in the TPG backup register for session T_{i+1} before shifting out the last response of session T_i . At this point the TPG backup register for session T_i cannot yet be overwritten, because it may still be needed for a repetition of T_i . At the end of session T_i , the obtained signature Q_i in the MISR is compared with the correct signature S_i . In case of a mismatch, the test is repeated after restoring the initial states of the TPG and the MISR from the backup registers. The number of repetitions is limited by a user-defined parameter W . If there is still a signature mismatch after W repetitions, then either a permanent fault has been detected or the rate of temporary failures is unacceptably high. The test is stopped and the device is rejected. The diagram in Figure 2 summarizes this flow. The actual number of repetitions already performed for a session is denoted by k .

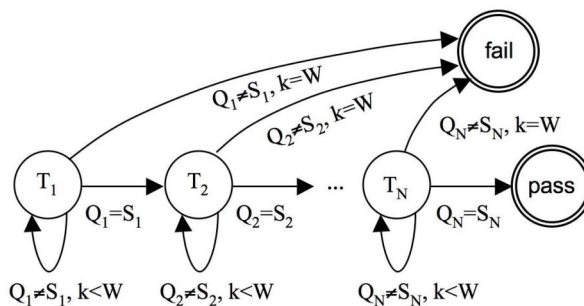


Figure 2: Test flow with rollback.

The hardware overhead for implementing the scheme is mainly determined by the backup registers and by the storage needed for the correct signatures S_1, \dots, S_N . As already indicated several solutions are possible for storing the signatures. They can be stored in extra registers or a small memory on chip, or they can be loaded from the ATE at the beginning of a test session.

For a given rate of temporary failures, the efficiency of the proposed scheme depends on the choice of the parameters W and N . As already pointed out above, the maximum number of iterations W reflects the acceptable error rate during system operation. If a frequent rollback during system operation is tolerable, then W can be increased. Selecting the appropriate number of

test sessions N is a more difficult task. On the one hand, a large value of N implies shorter sessions with a lower risk of temporary failures and helps to reduce yield loss. Furthermore, the time penalty is low when a short session has to be repeated. On the other hand, the parameter N determines the number of required reference signatures and thus the hardware overhead for an on-chip implementation. While yield improvement and minimization of hardware overhead are clearly contradictory goals, the more detailed analysis of the expected test time in the next section can help to find the best compromise.

3 An analytical model for test time

The problem of finding the number of test sessions minimizing the overall test time is similar to the problem of optimal checkpoint placement in classical fault tolerance [11]. However, the solutions from the literature cannot be directly applied, because for classical checkpoint placement the number of iterations is not limited. Therefore, in the sequel a specially tailored model for the expected overall test time with signature rollback is presented. To keep the analysis as simple as possible, the model developed in sections 3.1 and 3.2 assumes that no permanent faults are present in the circuit. The impact of permanent faults on the test time is discussed in Section 3.3. Furthermore, aliasing in the MISR is not taken into account, since shorter sessions also reduce the aliasing probability.

3.1 Duration of a test session with rollbacks

If no failure occurs during the test, partitioning the test into N sessions leads to the timing diagram in Figure 3, which also sets the lower bound on the overall time for a successful test.

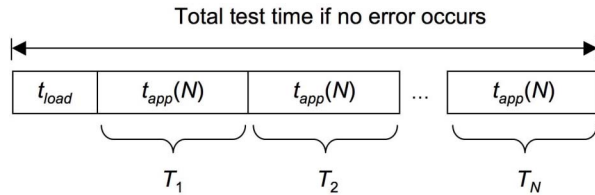


Figure 3: Minimum time for a successful test

After the first pattern has been loaded into the scan chains in t_{load} time units, the first session starts with the application of $x = \lceil X/N \rceil$ patterns. The test application time depends on N and is denoted by $t_{app}(N)$. As soon as the signature Q_i is available, it is compared against the reference signature, and the result is provided in the same clock cycle. At the end of a session the first pattern of the next session has already been loaded into

the scan chains, and the next session can immediately start with test application.

To determine the expected duration of a session including possible rollbacks in the presence of failures, let p denote the probability that at least one temporary failure occurs during a session. Accordingly, $1 - p$ is the probability that no temporary failure occurs. Assuming a constant rate λ of temporary failures, $1 - p$ can be determined using the exponential failure law from classical reliability theory [11], and the probabilities $1 - p$ and p are given by

$$1 - p = e^{-\lambda t_{app}(N)}, \quad p = 1 - e^{-\lambda t_{app}(N)}.$$

A session is executed exactly k times, $k < W$, if the first $k - 1$ iterations indicate a failure and the k -th iteration does not reveal any failure. The probability for this constellation is $p^{k-1}(1 - p)$. Since the number of repetitions is bounded by W , a session is executed exactly W times, if the first $W - 1$ iterations indicate a failure independent of the result of the W -th iteration. The probability for W iterations is thus given by p^{W-1} .

To determine the time for k iterations, it is necessary to distinguish two cases as illustrated in Figure 4.

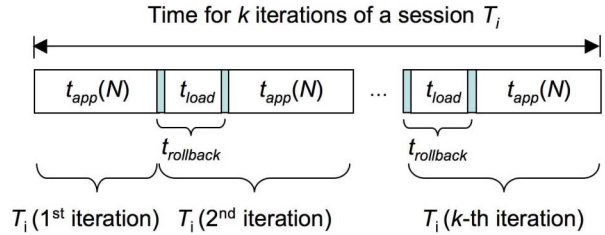


Figure 4: Time for k iterations of a session

If a session is executed for the first time, then the scan chains already contain the first pattern at the beginning of the session, and the time needed for the session is $t_{app}(N)$. After a rollback, the initial state of the test pattern generator must be loaded from the backup registers, and since the scan chains contain the first pattern of the next session, extra time is needed to shift in the first pattern again. As soon as the scan chains are completely loaded, the contents of the MISR can be restored from the backup register, and test application can be started. Independent of N , the time penalty $t_{rollback}$ for the rollback is therefore mainly determined by the length of the scan chains, and the overall duration of a repeated session is $t_{repeat}(N) = t_{app}(N) + t_{rollback}$. Consequently, the time for k iterations of a session is given by $t_{app}(N) + (k - 1) \cdot t_{repeat}(N) = k \cdot t_{app}(N) + (k - 1) \cdot t_{rollback}$. This results in the following equation for the expected duration of a test session with rollbacks.

$$\begin{aligned}
E(t_{sess}(N)) &= \sum_{k=1}^{W-1} (k \cdot t_{app}(N) + (k-1) \cdot t_{rollback}) p^{k-1} (1-p) \\
&\quad + (W \cdot t_{app}(N) + (W-1) \cdot t_{rollback}) p^{W-1}.
\end{aligned}$$

Elementary formula manipulations for geometric series provide the following expression for $E(t_{sess}(N))$.

$$E(t_{sess}(N)) = t_{repeat}(N) \cdot \frac{1-p^W}{1-p} - t_{rollback}.$$

This formula confirms the intuitive conjecture that shorter sessions lead to fewer failures and less penalty for iterations. The probability $p = 1 - e^{-\lambda_{app}(N)}$ of a temporary failure decreases with increasing N , and $(1-p^W)/(1-p)$ decreases with decreasing p .

3.2 Expected overall test time

For the test flow explained in Section 2, the overall test time depends on the duration of single test sessions and on the number of test sessions that are executed before the test is stopped. As explained in Section 2, the test is stopped, if the W -th iteration of a test session still results in a faulty signature. The probability for this event is p^W . Let $q = 1 - p^W$ denote the probability that the test is continued. Then the probability that the test is stopped after exactly i sessions, $i < N$, is given by $q^{i-1}(1-q)$. Furthermore, the probability that the test stops after exactly N sessions is q^{N-1} . Using the result derived in Section 3.1 for the expected duration of a test session with rollback, the expected overall test time $E(t_{total}(N))$ can be calculated as shown below.

$$\begin{aligned}
E(t_{total}(N)) &= t_{load} \\
&\quad + \sum_{i=1}^{N-1} i \cdot E(t_{sess}(N)) q^{i-1} (1-q) + N \cdot E(t_{sess}(N)) q^{N-1} \\
&= t_{load} + E(t_{sess}(N)) \cdot \frac{1-q^N}{1-q}.
\end{aligned}$$

Figure 5 shows the evolution of $E(t_{total}(N))$ for a test of the NXP circuit p951k applying 10,000 patterns at a frequency of 20 MHz [8]. The circuit contains 82 scan chains of maximum length 1122, the maximum number of repetitions has been set to $W = 2$, the number of test session varies between 1 and 100, and the failure rates range between 10^{-5} and 10^{-1} failures per millisecond. The curve for $\lambda = 10^{-5}$ shows the minimum test time for the fault free case for all values of N , and for $\lambda = 10^{-3}$ this ideal value is already reached for less than 20 test sessions. For $\lambda = 10^{-2}$ the probability that two iterations of a session fail is already very high and abortions of the test are very likely for small values of

N . Therefore the minimum test time is lower than for $\lambda = 10^{-5}$ and for $\lambda = 10^{-3}$. But with increasing N the probability of aborting the test is decreasing, which explains the increase of the total test time for a growing number of sessions. For $\lambda = 10^{-1}$ even a value of $N = 100$ is not sufficient to prevent yield loss.

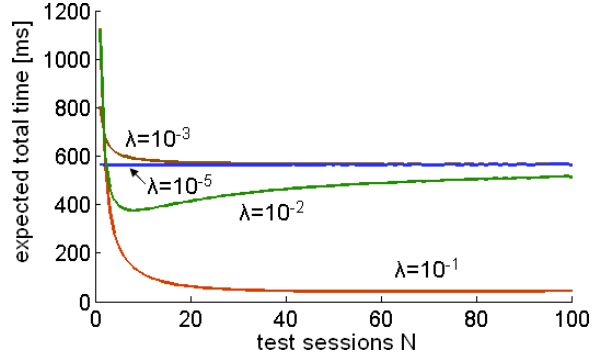


Figure 5: Test time as function of N ($W=2$)

The yield improvement can also be characterized by analyzing the probability q^N that a test is successfully completed for varying N . Figure 6 shows the results for the same scenario as investigated before.

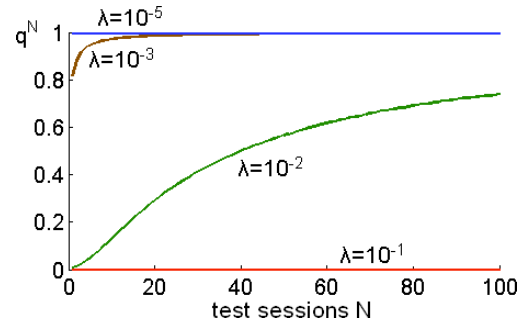


Figure 6: Probability of successfully completing the test ($W=2$)

The curves confirm the interpretations given for the test times in Figure 5. In addition, they also show, that mainly two different cases have to be considered when selecting the parameters for signature rollback. If the failure rate is in a range such that yield loss due to temporary failures is possible, but the failure rate is still relatively low ($\lambda = 10^{-3}$ in the example), then N can be selected, such that the test time is minimized. If the failure rate is higher, but yield improvement still seems possible ($\lambda = 10^{-2}$ in the example), then tuning the scheme just by increasing N would lead to a very high hardware overhead. In this case, increasing the number of repetitions W can help to speed up the convergence of the probability q^N to 1. This is demonstrated in Figure 7 and 8 by increasing W from $W = 2$ to $W = 4$.

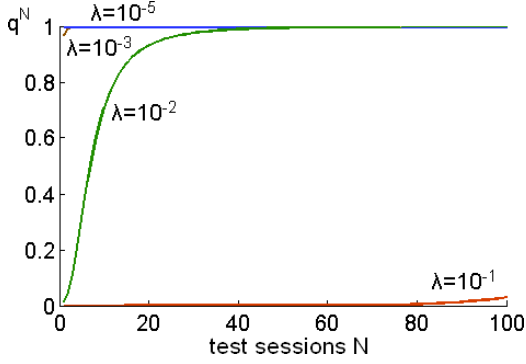


Figure 7: Probability of successfully completing the test ($W=4$)

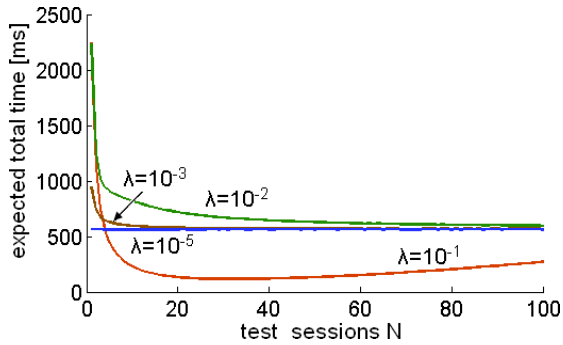


Figure 8: Test time as function of N ($W=4$)

Now the probability for a successful completion of the test reaches a very high level already for values around $N = 30$. In this region the test time also reaches its minimum.

3.3 Impact of permanent faults

If a permanent fault is present in the circuit, then in most cases the test time will even decrease, because the test is stopped after the session detecting the fault for the first time. In the worst case, the permanent fault appears in the last session of a test and there are no temporary failures in this session. In this case the time penalty for repeating this session W times is added.

Consequently, although the actual test times are slightly different in the presence of permanent faults, the analytical model presented in sections 3.1 and 3.2 still provides valid guidelines to find the best trade-off between hardware overhead, yield improvement and test time.

4 Experimental validation

As pointed out in Section 3, the analytical model for the expected overall test time need not take into account possible aliasing in the MISR. To validate the applicability of the model, a series of random experi-

ments has been performed. The experiments described below assume the scenario reported in [8], where a deterministic test is embedded into 10,000 pseudo-random patterns for the hard to test ISCAS89 and ITC99 benchmarks as well as for some NXP circuits. In all experiments a frequency of 20 MHz was assumed for the scan clock. For a fixed number of patterns and a fixed frequency of the scan clock, the expected test time mainly depends on the lengths of the scan chains, and circuits with similar scan chain lengths show similar results. Therefore only the results for the larger NXP circuits listed in Table 1 are presented in this section.

Table 1: Circuit Characteristics

Circuit Name	# Scan Elements	# Scan Chains	Length of a scan chain
p330k	18010	64	282
p418k	30430	64	476
p951k	91994	82	1122

For each circuit and each parameter combination a random experiment simulating the test was repeated 20 times and the mean value was compared to the result predicted by the analytical model. In each random experiment 10,000 test responses were randomly generated and errors were injected according to the failure rate λ under consideration. The results in Table 2 show that the simulation results closely match the prediction by the analytical model.

Table 2: Results for NXP circuits ($W=2$)

Circuit	N	λ	Analytical model	Simulation
p330k (141.5142)	1	10^{-1}	283.0282	283.0280
		10^{-3}	160.1862	169.8170
		10^{-5}	141.7143	155.6660
		10^{-7}	141.5162	141.5140
	10	10^{-1}	43.4021	43.8977
		10^{-3}	143.3773	142.9310
		10^{-5}	141.5346	143.6390
		10^{-7}	141.5148	142.9310
	30	10^{-1}	45.4516	42.2233
		10^{-3}	142.4232	141.9880
		10^{-5}	141.8053	142.2250
		10^{-7}	141.7987	141.9880
p418k (238.5240)	1	10^{-1}	477.0477	477.0480
		10^{-3}	289.1365	322.0070
		10^{-5}	239.0921	250.4500
		10^{-7}	238.5295	262.3760
	10	10^{-1}	55.2527	59.6764
		10^{-3}	243.5416	238.5340
		10^{-5}	238.5812	240.9120
		10^{-7}	238.5249	242.1070

Circuit	N	λ	Analytical model	Simulation
	30	10^{-1}	40.9890	46.5340
		10^{-3}	240.6843	240.9150
		10^{-5}	239.0214	240.1190
		10^{-7}	239.0025	240.5170
p951k (561.5560)	1	10^{-1}	1123.1000	1123.1100
		10^{-3}	802.8271	982.7230
		10^{-5}	564.7005	814.2560
		10^{-7}	561.5877	730.0230
	10	10^{-1}	113.0291	112.4120
		10^{-3}	584.3644	584.0590
		10^{-5}	561.8713	578.4190
		10^{-7}	561.5598	575.6080
	30	10^{-1}	48.4313	48.7383
		10^{-3}	570.3057	575.6230
		10^{-5}	562.7861	568.1220
		10^{-7}	562.6817	563.4330

For each circuit, the test time for the fault free case is attached to the circuit name to also show the evolution of test time with respect to this ideal case. Further experiments with varying parameter W showed the same trends, so that the analytical model presented in section 3 can be used as very accurate characterization of the test in the presence of temporary failures.

5 Conclusions

Static and dynamic parameter variations, device degradations and an increased susceptibility to soft errors make a robust design mandatory. Recent approaches efficiently implement time redundancy to cope with various types of delay errors and other timing problems. While these design efforts try to ensure a correct behavior in the presence of temporary failures, testing still has to address the circuit structure to identify permanent faults. The presented scheme for signature rollback targets an improved yield by distinguishing between critical permanent faults and non-critical transient failures. The analytical model for the expected overall test time accurately characterizes the test in the presence of temporary failures and provides the guidelines to find the best trade-off between hardware overhead, yield improvement and test time.

References

[1] P. H. Bardell and W. H. McAnney, "Self-Testing of Multichip Logic Modules," Proc. IEEE Int. Test Conf., Nov. 1982, pp. 200-204.
[2] C. Barnhart, et al., "OPMISR: The Foundation for Compressed ATPG Vectors," Proc. IEEE Int. Test Conf., Baltimore, MD, USA, Nov. 2001, pp. 748-757

[3] R. Baumann, "Soft errors in advanced computer systems," IEEE Design & Test of Computers, Vol. 22, No. 3, 2005, pp. 258-266.
[4] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," IEEE Micro, Nov.-Dec. 2005, pp. 10-16.
[5] M. A. Breuer, et al., "Defect and Error Tolerance in the Presence of Massive Numbers of Defects", IEEE Design & Test, Vol. 21, No. 3, May-June 2004, pp. 216-227.
[6] S. Das et al., "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction", IEEE Journal of Solid-State Circuits (JSSC), April 2006, pg. 792-804
[7] D. Ernst, et al., "Razor: Circuit-Level Correction of Timing Errors for Low Power Operation," IEEE Micro, Vol. 24, No. 6, Nov.-Dec. 2004, pp. 10-20.
[8] A.-W. Hakmi, et al., "Programmable Deterministic Built-in Self-test", Proc. IEEE Int. Test Conf., San Jose, CA, USA, Oct. 2007.
[9] S. Hellebrand, et al., "Built-in Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers", IEEE Trans. on Comp., Vol. 44, No. 2, Feb. 1995, pp. 223-233
[10] Z. Jiang and S. Gupta, "An ATPG for Threshold Testing: Obtaining Acceptable Yield in Future Processes", Proc. IEEE Int. Test Conf., Baltimore, MD, USA, 2002, pp. 824-833.
[11] I. Koren and C. M. Krishna, "Fault-Tolerant Systems," Morgan-Kaufman Publishers, San Francisco, CA, USA, 2007.
[12] S. Mitra, et al., "Robust System Design with Built-in Soft Error Resilience," IEEE Computer, Vol. 38, No. 2, Feb. 2005, pp. 43-52.
[13] S. Nassif, "Delay Variability: Sources, impact and trends," Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), 2000, pp. 368-369
[14] M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerant Circuits to Rescue Very Deep Submicron," Proc. 17th IEEE VLSI Test Symposium, April 1999.
[15] M. Nicolaidis, "GRAAL: A New Fault Tolerant Design Paradigm for Mitigating the Flaws of Deep Nanometric Designs," Proc. IEEE Int. Test Conf., San Jose, USA, Oct. 2007.
[16] I. Pomeranz and S. M. Reddy, "Generation of Functional Broadside Tests for Transition Faults," IEEE Trans. on CAD of Integrated Circuits and Systems, Oct. 2006, pp. 2207-2218.
[17] J. Rajski, et al., "Embedded Deterministic Test," IEEE Trans. on CAD of Circuits and Systems, Vol. 23, No. 5, May 2004, pp. 776-792.
[18] J. Rearick, "Too Much Delay Fault Coverage is a Bad Thing," Proc. IEEE Int. Test Conf., pp. 624-633, Oct. 2001.
[19] J. Saxena, et al., "A Case Study of IR-Drop in Structured At-Speed Testing," Proc. IEEE Int. Test Conf., 2003, pp. 1098-1104.
[20] S. Shahidi, S. K. Gupta, "ERTG: A Test Generator for Error-Rate Testing," Proc. IEEE Int. Test Conf., San Jose, USA, Oct. 2007.