Annales Univ. Sci. Budapest., Sect. Comp. 39 (2013) 161–180

SIGNATURE ROLLBACK WITH EXTREME COMPACTION – A TECHNIQUE FOR TESTING ROBUST VLSI CIRCUITS WITH REDUCED HARDWARE OVERHEAD

Thomas Indlekofer (Paderborn, Germany)

Dedicated to my father, Professor Karl-Heinz Indlekofer to his seventieth birthday

Communicated by Imre Kátai

(Received December 30, 2012; accepted February 06, 2013)

Abstract. With the decreasing feature size of today's nanoelectronic circuits, the susceptibility to transient failures increases. New robust and self-adaptive designs are developed, which can handle transient error to some extent, but at the same time make testing for permanent faults more difficult. This paper reviews the "signature rollback" scheme as a strategy to minimize both test time and yield loss. The main idea is to partition the test into shorter sessions and immediately repeat sessions with a faulty result to distinguish between permanent defects and transient failures. While a high number of test sessions leads to a high test quality, the hardware overhead also increases. For this, an extreme compaction method is added which reduces the amount of data to be stored on chip without any degradation of the product quality

1. Introduction

The number of transistors per chip has grown exponentially in the last decades and has brought tremendous progress in the performance and function-

Key words and phrases: BIST, Robust Design, Soft Errors, Time Redundancy, Overtesting.
The Project is supported by the DFG under contract HE 1686/3-2.

ality of today's semiconductor devices and microprocessors, like Gordon Moore postulated in 1965. While the first Intel microprocessor was developed 1971 with about 2200 transistors, in 2006 Intel announced its first billion-transistor Itanium microprocessor with about 1.72 billion transistors [21]. This is a direct result of the steadily decreasing feature sizes of transistors and interconnecting wires. As the physical implementation of Very Large Scale Integration (VLSI)-circuits is very complicated and passes through multiple production processes, the susceptibility to defects increases: already a small variation of geometrical shape can result in a permanent defect. In addition to that, chips are also affected by aging effects, external noise (soft errors) and parameter variations, which mostly result in transient faults and increase reliability problems [3], [5], [22]. To screen out chips with permanent defects, a manufacturing test is necessary which works as follows: every single chip is tested with an Automatic Test Equipment (ATE), which applies test patterns at the circuits' inputs. The test responses are compared to the reference responses. If the responses match the expected ones, the chip is assumed to be fault free. Otherwise, the chip will be rejected. The percentage of working chips is defined as yield and is a key indicator for semiconductor manufacturers. To maximize the product quality and to minimize yield loss by falsely rejected chips, test must be as accurate as possible.

To deal with transient faults, new robust and self-adaptive architectures have been developed which can compensate errors during system operation [11], [27]. One prominent example is the Razor register, which uses dynamic detection and correction of circuit timing errors to tune processor supply voltage and eliminate the need for voltage margins [9], [11]. The problem in testing robust designs is two-fold. On the one hand, observing only the input/ output behavior may provide a too optimistic quality assessment, because critical defects can be masked by the built-in redundancy. On the other hand, state of the art test procedures exploiting structural information cannot deal with the special characteristics of a robust design. Chips may be rejected during manufacturing test, even if the test reveals only non-critical failures that could be compensated during system operation. This problem of overtesting has already been addressed in the context of delay testing [6], [16], [24], [25], [26], [30]. In contrast to that, the BIST scheme presented in [2] 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. To reduce the necessary hardware overhead, a new extreme compaction scheme for response data has been introduced in [15]. This paper describes the resulting test scheme as an integral approach.

In the following Sections 2 and 3 the basic concepts of VLSI test, built-in self-test (BIST) and signature rollback will be summarized. Section 4 presents an analytical model for test time and quality, which can help to tune the

parameters of the test. The extreme compaction scheme will be introduced in Section 5, and its effect on yield improvement will be validated both analytically and by simulation experiments in section 6.

2. Background

This work focuses on digital circuits, which can be divided into two groups: combinational and sequential circuits. A combinational circuit implements the function

$$f_c : I \to O,$$

whereby $I = \{0,1\}^n$ is the set of inputs, and $O = \{0,1\}^m$ denotes the set of outputs. The behavior of a sequential circuit also depends on its internal state. It can be represented by a function

$$f_s : I \times F \to F' \times O,$$

whereby $F, F' = \{0,1\}^s$ describe the present and the next states of the circuits. The state of a circuit is stored in registers composed of flip-flops. Because the flip-flops are not directly accessible, the state of a sequential circuit can nether be directly controlled nor observed, which makes testing very difficult. To improve the testability of a sequential circuit, the design is converted into a so-called scan-design. Some extra logic is added, such that during test, the flip-flops can be configured into one or several scan-chains. Through extra scan-in and scan-out pins, test patterns can be shifted into the scan-chains, and the test responses can be shifted out. This way, the test of a sequential circuit is reduced to the test of a combinational circuit. Scan design is widely used in industry to simplify the test.

The goal of a manufacturing test of VLSI circuits is the detection of defects. The chip is tested with a set of test patterns $\mathcal{P} \subset \{0,1\}^n$, applied to the n inputs of the circuit-under-test (CUT). The respective responses $\mathcal{R} \subset \{0,1\}^m$ are observed at the CUT's outputs and compared to the predetermined responses $\mathcal{G} \subset \{0,1\}^m$. The test patterns are generated randomly or by software tools, the expected responses are obtained by simulation.

If for at least one pattern $P \in \mathcal{P}$ a mismatch $R \neq G$ occurs for the respective response, the chip is rejected because the CUT does not work as specified and failed the test. If for all patterns $P \in \mathcal{P}$ a match $R = G$ is observed, the CUT passes the test and can be further processed. With the growing complexity and heterogeneity of chips, test has become extremely challenging. Built-in self-test addresses these problems incorporating the test equipment into the chip itself

[1], [8], [19], [20]. For circuits with scan design the STUMPS (Self-test Using MISR and Parallel Shift Register Sequence Generator) architecture shown in figure 1 is widely used [4]. The test pattern generator (TPG) produces test patterns, which are shifted into the scan design. When all the scan chains are filled, the circuit works in normal mode for one clock cycle and the response is captured in the scan chains. While shifting out the responses and processing them with the multiple input signature register (MISR), the next pattern is shifted in.

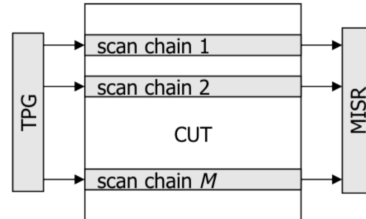

Figure 1: STUMPS Architecture

2.1. Test Pattern Generator (TPG)

The test pattern generator (TPG) is often based on linear feedback shift registers (LFSR), which consist of flip-flops and a selected number of exclusive-OR (XOR) gates [29]. A k-stage LFSR is shown in Figure 2.

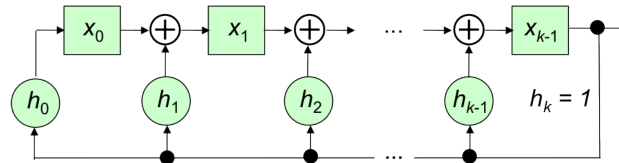

Figure 2: k-stage LFSR

The structure can be described by a characteristic polynomial $h(X) \in$ $\in GF(2)[X]$ of degree k

$$h(X) = X^k + \sum_{i=0}^{k-1} h_i X^i,$$

whereby $h_i = 1$, if there is a feedback path, and $h_i = 0$ else. The state transitions of the LFSR can also be described by a matrix $H \in M(k \times k, GF(2))$. With the initial state x, the states of the LFSR provide the sequence

$$x,\ Hx,\ H^2x,\ H^3x,\ \ldots .$$

The period of the sequence depends on the characteristic polynomial $h(X)$ and can reach $2^k - 1$ (all possible states without the all zero state). In this case, the polynomial has to be primitive [18].

Example. For $k = 4$, the polynomial $X^4 + X + 1$ is primitive. If an initial state $x \neq 0$ is selected, the state transition sequence has period $2^4 - 1 = 15$. If $x = 0$ is selected, all the following states will also be zero.

2.2. Test Response Evaluation

Storing the expected responses $\mathcal{G}$ on chip would require an enormous amount of extra memory. Therefore the responses must be compacted, such that only a few bits characterize $\mathcal{R}$. Signature analysis is the most popular response compaction technique used today [29]. It compacts the output responses into a signature and compares it with a golden signature for the fault-free case embedded on-chip. This time-compaction technique is based on polynomial division and implemented with an extended LFSR. For the sake of simplicity, the basic principle is explained here only for a circuit with one output producing a response sequence $R = (R_l, ..., R_0)$. The response sequence is combined with an LFSR with feedback polynomial $h(X) \in GF(2)[X]$. Interpreting R as a polynomial

$$R(X) = R_l X^l + \cdots + R_1 X + R_0 \in GF(2)[X],$$

the LFSR operation performs a polynomial division $R(X)/h(X)$. When $R(X)$ is shifted in completely, the state register contains the coefficients of the remainder polynomial $r(X) = r_{k-1}X^{k-1} + \cdots + r_1 X + r_0$ [23]. The vector $(r_0, ..., r_{k-1}) \in GF(2)^k$ is called the signature of the test. For the general case, the LFSR is extended to process several output streams in parallel, which leads to MISR. Using only signatures for response analysis can lead to aliasing, i.e. a faulty response sequence can be mapped to the same signature as the correct response sequence. Fortunately, the probability of aliasing is rather low. For a k- bit MISR it can be approximated by $P_{aliasing} \approx 2^{-k}$, which means that the quality of the compaction scheme mainly depends on the size of the MISR.

Example. Testing a circuit, which has 32 outputs, with 10,000 test patterns and using a 32 bit MISR for output compaction, the compaction ratio equals 10,000X and the aliasing probability $P_{aliasing} \approx 2^{-32} \approx 2,33 \cdot 10^{-10}$.

Describing this behavior with linear transformations by a Matrix H, a response sequence $R_l, \ldots, R_0$ and an initial state x lead to a signature sequence $x, Hx + R_l, H^2x + HR_l + R_{l-1}, \ldots, H^{l+1}x + H^l R_l + \cdots + HR_1 + R_0$, and $H^{l+1} + H^l R_l + \cdots + R_0$ is the final signature.

2.3. Test in the presence of transient failures

While permanent defects are problems for the semiconductor manufacturer, temporary transient failures due to external noises can be compensated during system operation e.g. with hardware redundancy. If such temporary failures occur during the manufacturing test, these failures have to be distinguishable from the permanent defects. A straightforward technique is to repeat tests with faulty outcomes. In case of a permanent defect the same faulty result can be observed again, whereas for a temporary failure the second test is likely to produce a different result. But 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 rejecting acceptable devices still remains a problem. Both solutions are not satisfactory according to the increasing susceptibility to temporary faults and the modern robust designs. In the next section, the signature rollback scheme is described, which addresses these problems by partitioning the test into several sessions.

3. Signature rollback

To distinguish between permanent and temporary failures, the technique proposed in [2] 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 the already discussed STUMPS architecture and is shown in Figure 3. But the idea works also for a test with deterministic test patterns, as described in [14].

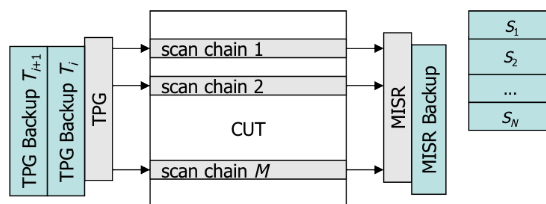

Figure 3: Signature Rollback Architecture

To realize the test with rollback, a given test $\mathcal{T}$ with X patterns is partitioned into N sessions $T_1, \ldots, T_N$, 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 by storing it on chip. 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 4 summarizes this flow. The actual number of repetitions already performed for a session is denoted by w.

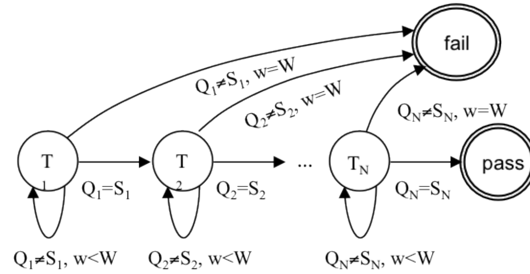

Figure 4: Test flow with rollback

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.

4. An analytical model for the test time and yield improvement

The problem of finding the number of test sessions improving the overall test time and yield is similar to the problem of optimal checkpoint placement in classical fault tolerance [17]. However, 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 and yield improvement is presented. To keep the analysis as simple as possible, the model developed in the next subsections assumes that no permanent faults are present in the circuit. The impact of permanent faults on the test time and yield is discussed in Section 4.3. Furthermore, aliasing in the MISR is not taken into account, since shorter sessions also reduce the aliasing probability.

4.1. 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 5.

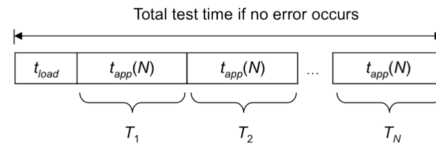

Figure 5: Minimum time for a successful test

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 [17], and the probabilities $1 - p$ and p are given by

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

A session is executed exactly w times, $w < W$, if the first $w-1$ iterations indicate a failure and the w-th iteration does not reveal any failure. The probability for this constellation is $p^{w-1}(1-p)$. 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 w iterations, it is necessary to distinguish two cases as illustrated in Figure 6.

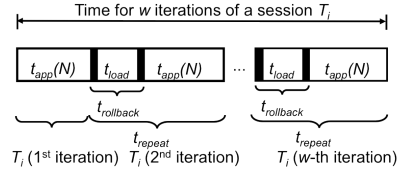

Figure 6: Time for w iterations of a session

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 w iterations of a session is given by $t_{app}(N) + (w-1) \cdot t_{repeat}(N)$. This results in the following equation for the expected duration of a test session with rollbacks.

$$E\left(t_{sess}(N)\right) = \sum_{w=1}^{W-1} (w \cdot t_{app}(N) + (w-1) \cdot t_{rollback}) p^{w-1}(1-p) + \\ + (W \cdot t_{app}(N) + (W-1) \cdot t_{rollback}) p^{W-1}. \tag{4.1}$$

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

$$E\left(t_{sess}(N)\right) = t_{repeat}(N) \cdot \frac{1-p^W}{1-p} - t_{rollback}. \tag{4.2}$$

This formula confirms the intuitive conjecture that shorter sessions lead to fewer failures and less penalty for iterations. The probability $p = 1 - e^{-\lambda t_{app}(N)}$

of a temporary failure decreases with increasing N, and $\left(1-p^W\right)/(1-p)$ decreases with decreasing p.

4.2. Expected overall test time and yield improvement

For the test flow explained in Section 3, 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 3, 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 4.1 for the expected duration of a test session with rollback, the expected overall test time $E\left(t_{total}(N)\right)$ can be calculated as shown below.

$$
\begin{aligned}
E\left(t_{total}(N)\right) &= t_{load} + \sum_{i=1}^{N-1} i \cdot E\left(t_{sess}(N)\right) q^{i-1}(1-q) + \\
&\quad + N \cdot E\left(t_{sess}(N)\right) q^{N-1} = \\
&= t_{load} + E\left(t_{sess}(N)\right) \cdot \frac{1-q^N}{1-q}.
\end{aligned}
\tag{4.3}
$$

Figure 7 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 [13]. 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 sessions 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.

The yield improvement can be characterized by analyzing the probability q^N that a test is successfully completed for varying N. Figure 8 shows the results for the same scenario as investigated before.

The diagram confirms the interpretations given for the test times in Figure 7. The curve for $\lambda = 10^{-5}$ already corresponds to the fault free case. For

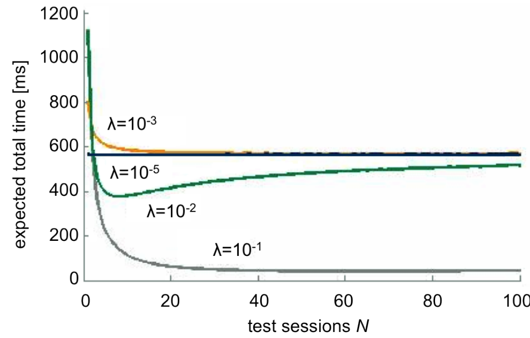

Figure 7: Test time as function of N(W=2)

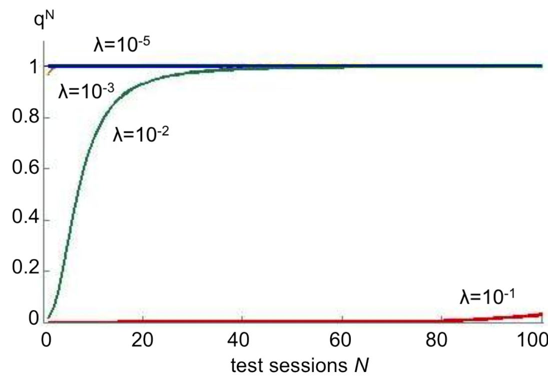

Figure 8: Probability of successfully completing the test (W=2)

the other failure rates it can be observed, that with increasing N the probability q^N increases, too. For $\lambda = 10^{-1}$ this effect is not yet observable even for a value of $N = 100$. N has to be further increased significantly to prevent yield loss.

4.3. 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. The product quality is as good as a standard test, because the signature of the last session is the "normal" signature of a standard test. Consequently, although the

actual test times are slightly different in the presence of permanent faults, the analytical model presented in sections 4.1 and 4.2 still provides valid guidelines to find the best trade-off between hardware overhead, yield improvement and test time.

5. Compacting the Reference Signature

As illustrated by Figures 7 and 8, both the test time and the test quality are improved with increasing number of test sessions N. To reach this effect also for high failure rates, N has to be increased significantly. The problem is, that this requires to store an increasing amount of reference data on chip. To minimize the hardware overhead while maintaining the test quality, the method in [15] works with compacted signatures, but doesn't require additional test generation procedures like the methods in [10] and [28].

5.1. Architecture

To reduce the amount of reference data, the signatures are compacted as shown in Figure 9. Instead of storing the complete k- bit signatures

$$S_i = (S_{i0}, \ldots, S_{ik-1}) \in GF(2)^k,$$

only the parities $\Pi(S_i) := \sum_{j=0}^{k-1} S_{ij}$ are stored. The actual parities $\Pi(Q_i)$ obtained at the end of the sessions are compared to the reference parities $\Pi(S_i)$.

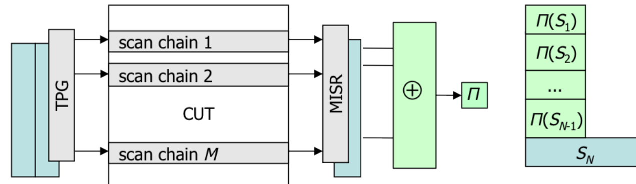

Figure 9: Architecture for embedded test with extreme compaction.

Due to the compaction some errors may be masked, although they could be observed within the MISR. Therefore, the following cases have to be distinguished: if the complete reference signature for the last session, i.e. for the whole test T, is still stored, then the fault coverage for permanent faults will be as high as for a standard BIST without signature rollback. If due to the compaction transient faults are masked during the whole test, the number of

repetitions will be reduced. But if transient faults are masked in one session and detected with certain latency in one of the following sessions, the yield gain of the original technique will be reduced. Figure 10 shows an example. Here, a transient fault in session T_i leads to a deviation of the actual signature from the reference signature, i.e. $Q_i \neq S_i$. Since $\Pi(Q_i) = \Pi(S_i)$ holds for the respective parity bits, the error is not detected. The test continues and session T_{i+1} is started with a wrong seed for the MISR. This leads to a mismatch $Q_{i+1} \neq S_{i+1}$ after session T_{i+1}, but this time also $\Pi(Q_{i+1})$ and $\Pi(S_{i+1})$ differ. As the initial MISR state for session T_{i+1} is incorrect, the chip will be rejected independent of the number of repetitions. At this point it should be noted that a standard test without rollback would also reject the chip because of the faulty signature caused by the transient fault. Overall, the yield for signature rollback with additional compaction is still higher than that for a standard test, but the benefits of the rollback technique cannot be fully exploited.

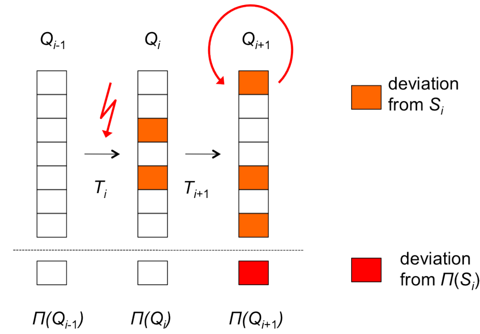

Figure 10: Aborted test due to fault detection latency.

For maximizing the yield gain, a method is needed which guarantees the lowest possible error latency for transient faults. To reach this goal, a sequence of consecutive parity bits of length L will be monitored at the end of each session (cf. Figure 11). Accordingly, L parity bits per session will be stored.

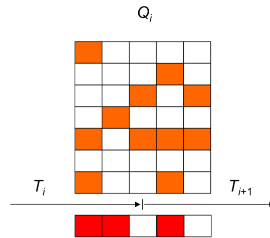

Figure 11: Parity window around the signature Q_i.

To accomplish this, the proposed architecture of Figure 9 has to be slightly extended. Instead of using a single flip-flop to store the parity bit, a shift-register is needed to accommodate the L-bit parity sequence. The necessary size L of the parity window can be determined with the help of the behavior of a MISR shown in section 5.2.

5.2. Error propagation in a MISR

Observation 5.1. *Consider a MISR with a primitive feedback-polynomial $h(X) \in GF(2)[X]$. Assume, that at time $t = i$ a possible error impacts an odd (even) number of bits in the MISR. Then the probability, that the error always impacts an odd (even) number of bits up to the time $t = i + L$, can be estimated by 2^{-L}.*

Proof. According to the matrix representation of a MISR introduced in Section 2.2, in case of an error at time $t = i$ the vector $R_{l-i} + e$ instead of R_{l-i} enters the MISR. It is easy to check, that the following states differ from the fault free states by $He, H^2e, H^3e, .$ This implies that error propagation behaves like the state sequence of an LFSR with feedback-polynomial $h(X)$ and initial state e. Since $h(X)$ is primitive, an odd number of feedback-coefficients must be nonzero. If the state bit x_{k-1} equals 1, an odd number of bits are inverted and so the parity changes. Furthermore, as the LFSR has the properties of a pseudo-random generator in autonomous mode, the probability of a 1 at x_{k-1} is approximately 2^{-1}. Accordingly, the probability, that an error always impacts an odd (even) number of bits, can be estimated by 2^{-L}. ■

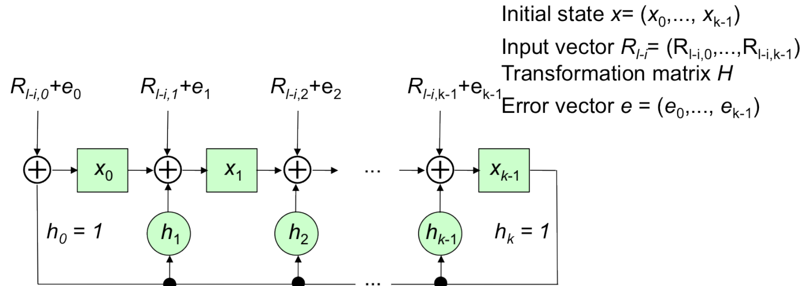

Figure 12: MISR

As a result of the theorem, observing a parity sequence of length L at the end of each session reduces the probability of error masking to 2^{-L}. Accordingly, the probability of error detection is given by $1-2^{-L}$. For $L = 8$ the probability of error propagation is already lower than 0.01. So the reduction in yield improvement compared to the original rollback scheme is less than 1 %. In the

case of a 64-bit MISR the data volume is reduced to one eighth. For higher MISR lengths the reduction can be even better. This estimation also holds, if an error in the circuit influences several scan elements and erroneous data enter the MISR at several points in time. In this case it can be assumed without loss of generality, that $t = i$ is the last point in time, at which the MISR receives an erroneous input.

6. Experiments

To validate the method proposed in Section 5.1-5.2 a series of experiments for ISCAS'89 [7], ITC'99 [12] and NXP benchmark circuits has been performed. The relevant characteristics of the examined benchmark circuits are listed in Table 1.

Circuit	(P)PI	(P)PO	FF	Scan chains	Max. length
s13207.1	700	790	638	10	79
s15850.1	611	684	534	18	41
s35932	1763	2048	1728	32	64
s38417	1664	1742	1636	26	67
s38584	1464	1730	1426	26	67
b17	1452	1512	1415	18	84
b22	767	757	735	12	64
p330k	18010	17468	17226	64	282
p388k	25005	24065	24065	50	569
p418k	30430	29809	29205	64	664
p469k	635	403	332	1	635
p483k	33264	32610	32409	71	469
p500k	30768	30840	30731	76	406
p533k	33373	32610	32409	71	471
p874k	61977	70863	42076	59	1202
p951k	91994	104714	104624	82	1277

Table 1: Characteristics of the examined circuits.

The first series of experiments examined the error detection latencies for transient faults for an extreme compaction into a single parity bit. A test with 10,000 random patterns was partitioned into 10 sessions of equal length. A single transient fault was randomly injected into the circuit during the first session. The test was simulated until the first deviation from the expected parity $\Pi(Q_i) \neq \Pi(S_i)$ was detected in session T_i, i.e. with latency $i-1$, or the test was successfully completed, which corresponds to an infinite error detection latency. Each experiment was repeated 100 times. Table 2 lists the

frequency distribution of the error detection latencies L_D. The values confirm the problems pointed out above with respect to the extreme compaction into a single parity bit. Even for the best case (s13207.1) the error was detected immediately only in 75 % of the experiments. In 25 % of the experiments the error was detected with latency L_D, $0 < L_D < \infty$. In these cases the test was aborted unnecessarily. In the worst case (p330k), the values are shifting to 26% for immediate detection and 74% for unnecessary test aborts.

Circuit	Number of experiments with detection latency L_D					
	$L_D = 0$	$L_D = 1$	$L_D = 2$	$L_D = 3$	$L_D = 4$	$L_D = \infty$
s13207.1	75	15	0	10	0	0
s15850.1	31	37	1	16	1	0
s35932	31	37	1	16	1	0
s38417	46	16	17	21	0	0
s38584	46	16	17	21	0	0
b17	45	33	15	7	0	0
b22	48	32	9	3	8	0
p330k	26	55	3	4	4	0
p388k	33	17	11	25	1	0
p418k	33	17	11	25	1	0
p469k	61	22	12	1	3	0
p483k	61	22	12	1	3	0
p500k	70	13	8	8	1	0
p533k	59	26	11	2	0	0
p874k	50	37	6	0	1	0
p951k	53	43	3	1	0	0

Table 2: Frequency distribution of error-latency for 1 parity bit.

The second series of experiments validated the estimation of error propagation inside the MISR. Again, a transient fault was randomly injected into the circuit during the first session. But this time the parities were monitored continuously. For every MISR-cycle the actually computed and the reference parities were compared. If the parities are indistinguishable for L MISR-cycles, this will be called aliasing-sequence of length L. In Table 3 the frequency distributions of aliasing-sequences are shown. The values represent the average percentage of aliasing-sequences of length L within the whole set of aliasing-sequences for 10 experiments per benchmark. It can be seen, that the values are very close to the predicted trend 2^{-L}. Finally, the error detection latency was analyzed for the proposed method with respect to the size of the parity window. For this, again a transient fault was randomly injected into the circuit during the first session. As in the first series of experiments the MISR was monitored during the whole test. Furthermore the parity sequences were calculated, and at the end of each session the computed parity sequence was

compared to the reference sequence. Once the error was detected, the error latency was determined and the test was aborted. Table 4 shows the summary of all results. For the sake of clarity the obtained data are consolidated. For every parity window size only the number of critical cases, where $0 < L_D < \infty$, is shown. The results show, that for $L \geq 7$ only a few experiments yield a critical error latency resulting in unnecessary test aborts. Again these results confirm the estimations made in Section 5.1-5.2.

Circuit	Aliasing sequences: Proportion of sequences of length L					
	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$
s13207.1	0.50006	0.25022	0.12472	0.06255	0.03114	0.01568
s15850.1	0.49877	0.2507	0.12513	0.06273	0.0313	0.01547
s35932	0.50018	0.24961	0.12498	0.0626	0.03133	0.01565
s38417	0.50038	0.24956	0.12498	0.06265	0.0313	0.0155
s38584	0.49986	0.24982	0.12517	0.06263	0.0315	0.01541
b17	0.50082	0.25005	0.12468	0.06217	0.03103	0.01559
b22	0.49999	0.24987	0.12508	0.06255	0.03105	0.01576
p330k	0.49914	0.25029	0.12578	0.0622	0.03132	0.01559
p388k	0.50051	0.2496	0.12494	0.06234	0.03127	0.01569
p418k	0.50114	0.24959	0.12429	0.06217	0.03144	0.01567
p469k	0.49944	0.25069	0.12488	0.06244	0.03126	0.01565
p483k	0.50038	0.24946	0.12499	0.06248	0.0314	0.01551
p500k	0.49923	0.25002	0.12521	0.06247	0.03175	0.01568
p533k	0.49939	0.24971	0.12531	0.06258	0.03164	0.01567
p874k	0.50083	0.24983	0.1247	0.06206	0.03109	0.01556
p951k	0.49983	0.24951	0.12533	0.06305	0.03098	0.01554

Table 3: Aliasing sequences

Assuming a parity window of size $L = 8$, the possible reduction of the storage requirements for the reference data depends on the MISR-size k. The storage reduction can be calculated as $\frac{k \times N}{8 \times N} = \frac{k}{8}$ and is between 4 and 12 in our experiments. The additional hardware consists of only one XOR-tree to generate the parities and one additional L bit register for accommodating the actually computed parities. The size of the XOR-tree depends on the MISR-size ($k - 1$ XOR-gates).

7. Conclusion

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

Circuit	Number of Experiments with Latency $0 < L_D < \infty$ using a parity-window with L bits							
	$L=1$	$L=2$	$L=3$	$L=4$	$L=5$	$L=6$	$L=7$	$L=8$
s13207.1	25	9	7	3	0	1	0	0
s15850.1	69	48	39	24	23	1	0	0
s35932	58	27	15	0	0	0	0	0
s38417	54	26	8	8	9	0	0	0
s38584	42	27	37	20	0	3	0	0
b17	55	40	18	12	8	1	0	0
b22	52	26	6	3	1	0	0	0
p330k	74	35	7	8	4	0	0	0
p388k	67	9	4	0	0	0	0	0
p418k	63	28	30	1	0	0	0	0
p469k	39	17	5	5	1	3	1	0
p483k	66	26	16	15	13	7	0	0
p500k	30	20	5	2	1	0	0	0
p533k	41	19	6	0	0	0	0	0
p874k	50	29	29	0	0	0	2	0
p951k	47	19	19	3	5	0	0	0

Table 4: Error latency depending on the number of parity bits

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 presented scheme also reduces the storage amount of the necessary reference data.

Acknowledgement. I would gratefully acknowledge the contribution of my supervisor Prof. Dr. Sybille Hellebrand, head of the Computer Engineering Group at the University of Paderborn, for her very helpfully guidance, encouragement and excellent advice in working at this topic and writing this paper.

References

[1] **Abramovici, M., M.A. Breuer, and A.D. Friedman,** *Digital Systems Testing and Testable Design*, IEEE Press, Piscataway, NJ, 1994 (revised printing).

[2] **Amgalan, U., C. Hachmann, S. Hellebrand, and H.-J. Wunderlich,** *Signature Rollback A Technique for Testing Robust Circuits*, Proc. IEEE VLSI Test Symposium, San Diego, CA, USA, April 2008, pp. 125–130.

[3] **Baumann, R.,** *Soft errors in advanced computer systems*, IEEE Design & Test of Computers, Vol. 22, No. 3, 2005, pp. 258–266.

[4] **Bardell, P.H. and W.H. McAnney,** *Self-Testing of Multichip Logic Modules*, Proc. IEEE Int. Test Conf. (ITC82), Nov. 1982, pp. 200–204.

[5] **Borkar, S.,** *Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation*, IEEE Micro, Nov.-Dec. 2005, pp. 10–16.

[6] **Breuer, M.A., S.K. Gupta, and T.M. Mak,** *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.

[7] **Brglez, F., D. Bryan, and K. Kozminski,** *Combinational Profiles of Sequential Benchmark Circuits*, Proc. IEEE Int. Symp. on Circuits and Systems, 1989, pp. 1929–1934.

[8] **Bushnell, M.L. and V.D. Agrawal,** *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Springer Science, New York, 2000.

[9] **Das, S. et al.,** *A Self-Tuning DVS Processor Using Delay-Error Detection and Correction*, IEEE Journal of Solid-State Circuits (JSSC), April 2006, pp. 792–804.

[10] **Elm, M. and H.-J. Wunderlich,** *BISD: Scan-Based Built-In Self-Diagnosis*, Proc. Design Automation and Test in Europe (DATE'10), Dresden, Germany, March 8–12, 2010.

[11] **Ernst, D. 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.

[12] Benchmark information and circuits, available at `http://www.cerc.utexas.edu/itc99-benchmarks/bench.html`

[13] **Hakmi, A.-W. et al.,** *Programmable Deterministic Built-In Self-Test*, Proc. IEEE Int. Test Conf. (ITC'07), San Jose, CA, USA, Oct. 2007.

[14] **Hellebrand, S. 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.

[15] **Indlekofer, T., M. Schnittger, and S. Hellebrand,** *Efficient Test Response Compaction for Robust BIST Using Parity Sequences*, Proc. 28th IEEE Int. Conference on Computer Design (ICCD'10), Amsterdam, The Netherlands, October 2010, pp. 480-485.

[16] **Jiang, Z. and S. Gupta,** *An ATPG for Threshold Testing: Obtaining Acceptable Yield in Future Processes*, Proc. IEEE Int. Test Conf. (ITC'02), Baltimore, MD, USA, 2002, pp. 824-833.

[17] **Koren, I. and C.M. Krishna,** *Fault-Tolerant Systems*, Morgan-Kaufman Publishers, San Francisco, CA, USA, 2007.

[18] **Lidl, R. and H. Niederreiter,** *Finite Fields*, Cambridge University Press, Cambridge, United Kingdom, 1997.

[19] **McCluskey, E.J.,** *Logic Design Principles: With Emphasis on Testable Semiconductor Circuits*, Prentice Hall, Englewood Cliffs, NJ, 1986.

[20] **Mourad, S. and Y. Zorian,** *Principles of Testing Electronic Systems*, John Wiley & Sons, Somerset, NJ, 2000.

[21] **Mukherjee, S.,** *Architecture Design for Soft Errors*, Morgan Kaufmann Publishersm Burlington, MA, USA, pp. 1–2.

[22] **Nassif, S.,** *Delay Variability: Sources, impact and trends*, Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), 2000, pp. 368–369.

[23] **Peterson, W.W. and E.J. Weldon, Jr.,** *Error-Correcting Codes*, MIT Press, Cambridge, MA, 1972.

[24] **Pomeranz, I. and S. M. Reddy,** *Generation of Functional Broadside Tests for Transition Faults*, IEEE Transactions on CAD, Oct. 2006, pp. 2207–2218.

[25] **Rearick, J.,** *Too much Delay Fault Coverage is a Bad Thing*, Proc. IEEE Int. Test Conf. (ITC'01), pp. 624- 633, Oct. 2001.

[26] **Shahidi, S. and S.K. Gupta,** *ERTG: A Test Generation for Error - Rate Testing*, Proc. IEEE Int. Test Conf. (ITC'07), San Jose, USA, Oct. 2007.

[27] **Sylvester, D., D. Blaauw, and E. Karl,** *ElastIC: An Adaptive Self-Healing Architecture for Unpredictable Silicon*, IEEE Design and Test, Vol. 23, No. 6, November–December 2006, pp. 484–490.

[28] **Vranken, H. et al.,** *Fault Detection and Diagnosis with Parity Trees for Space Compaction of Test Responses*, Proc. 43rd ACM/IEEE Design Automation Conf. (DAC'06), San Francisco, CA, USA, July 2006, pp. 1095–1098.

[29] **Wang, L.-T., C.-W. Wu and X. Wen,** *VLSI Test Principles and Architectures: Design for Testability*, Morgan-Kaufman Publishers, San Francisco, CA, USA, 2006.

[30] **Yuan, F. and Q. Xu,** *On Systematic Illegal State Identification for Pseudo- Functional Testing*, Proc. DEsign Automation Conference (DAC'09), San Francisco, CA, USA, July 26-31, 2009, pp. 702–707.

Thomas Indlekofer
University of Paderborn
Paderborn
Germany
thomas.indlekofer@uni-paderborn.de