Aging Monitor Reuse for Small Delay Fault Testing

Liu, Chang; Kochte, Michael A.; Wunderlich, Hans-Joachim

Proceedings of the 35th VLSI Test Symposium (VTS'17) Caesars Palace, Las Vegas, Nevada, USA, 9-12 April 2017

doi: http://dx.doi.org/10.1109/VTS.2017.7928921

Abstract: Small delay faults receive more and more attention, since they may indicate a circuit reliability marginality even if they do not violate the timing at the time of production. At-speed test and faster-than-at-speed test (FAST) are rather expensive tasks to test for such faults. The paper at hand avoids complex on-chip structures or expensive high-speed ATE for test response evaluation, if aging monitors which are integrated into the device under test anyway are reused. The main challenge in reusing aging monitors for FAST consists in possible false alerts at higher frequencies. While a certain test vector pair makes a delay fault observable at one monitor, it may also exceed the time slack in the fault free case at a different monitor which has to be masked. Therefore, a multidimensional optimizing problem has to be solved for minimizing the masking overhead and the number of test vectors while maximizing delay fault coverage.

Preprint

Aging Monitor Reuse for Small Delay Fault Testing

Chang Liu, Michael A. Kochte, and Hans-Joachim Wunderlich
ITI, University of Stuttgart, Pfaffenwaldring 47, D-70569, Stuttgart, Germany
Email: Chang.Liu@iti.uni-stuttgart.de, kochte@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de

***Abstract*—Small delay faults receive more and more attention, since they may indicate a circuit reliability marginality even if they do not violate the timing at the time of production. At-speed test and faster-than-at-speed test (FAST) are rather expensive tasks to test for such faults.**

The paper at hand avoids complex on-chip structures or expensive high-speed ATE for test response evaluation, if aging monitors which are integrated into the device under test anyway are reused. The main challenge in reusing aging monitors for FAST consists in possible false alerts at higher frequencies. While a certain test vector pair makes a delay fault observable at one monitor, it may also exceed the time slack in the fault free case at a different monitor which has to be masked. Therefore, a multidimensional optimizing problem has to be solved for minimizing the masking overhead and the number of test vectors while maximizing delay fault coverage.

***Keywords*-Delay monitoring, delay test, faster-than-at-speed test, stability checker, small delay fault, ATPG**

I. Introduction

Aggressive technology scaling increases the possibility of manufacturing defects, the susceptibility to aging and it aggravates process variations. Defects such as resistive opens, resistive bridges, gate-oxide defects or parametric deviations of transistors often manifest themselves as small delay faults (SDF), introducing an additional small delay at cells or interconnects [1], [2]. SDFs can also arise from power supply noise, crosstalk or aging degradation, intensified by process variation [3] [4]. In addition, low-power and near-threshold operation reduces the noise immunity and makes the circuit highly susceptive to small delay deviations. As a result, many test and diagnosis methods targeting SDFs have been proposed [4], based e.g. on pattern selection [5], pattern grouping [6], or timing-aware ATPG [7].

Hidden delay faults (HDFs) [8] are a subset of SDFs for which the slack of the longest sensitizable path through the fault site is larger than the fault size, thus the fault effect cannot be detected by at-speed or even timing-aware delay test. Although such HDFs do not violate the nominal timing, the marginal hardware may pose a reliability risk later due to aging and degradation in field [9], when for instance oxide defects grow or the delay of a resistive via magnifies due to electromigration. Thus, an early life failure (ELF) or a functional timing failure may occur after product shipment [10].

HDFs that degrade rapidly under stress can be detected by burn-in. Burn-in stresses the hardware, e.g. by increased voltage or temperature, to accelerate defect degradation and to detect imperfections and ”infant mortality” (ELFs). The effectiveness of burn-in test with increased voltage has been reduced exponentially due to voltage scaling [11]. Burn-in may also reduce the reliability of the devices [12]. Additionally, burn-in is very expensive due to the required equipment and high test time.

Faster-than-at-Speed test (FAST) applies test patterns at frequencies above the nominal speed to detect SDFs or HDFs [13, 14]. For high fault coverage, a high maximum test frequency and many different frequencies are often required. Dedicated and expensive automatic test equipment (ATE) is required for FAST to overcome parasitic capacitance effects and tester skew [14].

An alternative is to support FAST with on-chip clock generators [15], [16], test controllers and result evaluators. During FAST, some transitions propagated through long paths may not reach the outputs or pseudo-outputs in time and cause unknown values (X) in the test responses, which requires dedicated test control and response evaluation structures, e.g. based on multiplexers or an X-tolerant MISR and memory for intermediate signatures [8, 17]. Furthermore, IR-drop analysis and pattern reordering can be applied to avoid false positive detections due to IR-drop during FAST [6].

Aging mechanisms shift parameters in transistors and interconnects and increase their delay over the lifetime. For an in-field delay monitoring, different types of aging monitors have been developed and integrated in the circuits [18–22]. The paper at hand concentrates on in-situ delay monitors which measure the performance indicator directly from the actual paths of the circuit. The Razor flip-flops [23] detect and later correct the timing failure by comparing the values in original registers and shadow latches with a delayed clock. Delay detecting flip-flops with stability checker [24–27] or comparison logic [24, 28] sense the degradation progress and generate an aging alert when a transition of the observed signal violates a predefined time period (guard band) before the sampling time. To reduce the overhead, the monitors are often placed at terminals of the critical or long paths [25, 29]. SlackProbe [30] does not limit monitor placement to path-ends, but also allows intermediate placement to save hardware cost. Observation Point Selection [31] takes path sensitization into account when placing the monitors at meticulously selected positions in the circuit nets. The method improves the hardware efficiency and indicates the aging process earlier with more frequent slack measurements of path prefixes. Even though imminent failures can be predicted by such aging monitors, the test for HDFs and ELFs is still a necessity, since product quality needs to be evaluated to reduce costly field returns and to improve manufacturing processes.

Both aging and delay faults change the circuit path delay. We screen both issues by detecting the deviated delay with a uniform structure that reuses delay detecting aging monitors for hidden delay fault detection. This approach can achieve a high coverage of target faults with a few fixed test frequencies. Due to reuse, it induces very limited hardware overhead. Since transitions delayed by HDFs can be directly detected by delay monitors, extra instruments for result evaluation, e.g. an ATE or MISR, are not required. Also, X-tolerant structures [8, 17] can be avoided.

The contributions of this paper are:

- We propose a novel approach for HDFs detection by *reuse of aging monitors* for at-speed and faster than at-speed delay test.
- To prevent false alert of monitors during FAST, we introduce a monitor and test pattern selection scheme that tailors the test pattern set and selectively enables the monitors for each test frequency.
- To maximize fault coverage, we model the monitor and pattern selection as a Pseudo-Boolean optimization problem and solve it by a SAT-solver.

Section II gives an overview of the proposed monitor reuse approach. In Section III, the monitor and pattern selection is formulated as a Pseudo-Boolean optimization problem for fault coverage maximization. Finally in Section IV, the experimental setup and results are discussed.

II. Delay Monitor Reuse Overview

A. Delay Detecting Flip-flop

A delay detecting flip-flop is a standard flip-flop extended with a delay monitor [24]. The monitors sense the transitions during a predefined detection window and are often placed at the end of critical paths or selected intermediate positions of circuits. In the following discussion we assume monitors are integrated in the flip-flops at the end of long paths. The proposed monitor reuse approach can be also applied to intermediate node monitor placement.

A delay monitor may consist of a delay element, stability checker and latch (Fig. 1 (a)). An alternative design (Fig. 1 (b)) contains a delay element (with delay T_g), shadow register and an XOR gate. Because of the delay element, the observed signal *D* is *pre*sampled by the shadow register. If the value of *D* changes within T_g, different values are captured in the original and shadow registers and an timing alert is issued. The time period T_g right before the rising clock edge is defined as the transition detection window, i.e. the time during which the signal stability is checked. To distinguish the transition detection window during degradation measurement and during small delay test, we refer to the detection period as Guard Band for aging prediction and use the term detection window for the testing scenario. The output of the original (*Q*) and shadow register (*Q'*) are compared by the XOR gate. We choose the delay monitor in [28] based on the structure of Fig. 1 (b) in the following discussion and implementation.

If an observed signal D_{Nominal} reaches its stable value before the Guard Band, then the path is functional and uncritical. On the contrary, if a signal (D_{Aged}) is unstable during the Guard Band, i.e. D'_{Aged} (D_{Aged} delayed by T_g) arrives after the rising clock edge, then different values are compared by the XOR gate and an alert is generated (Fig. 1 (b)).

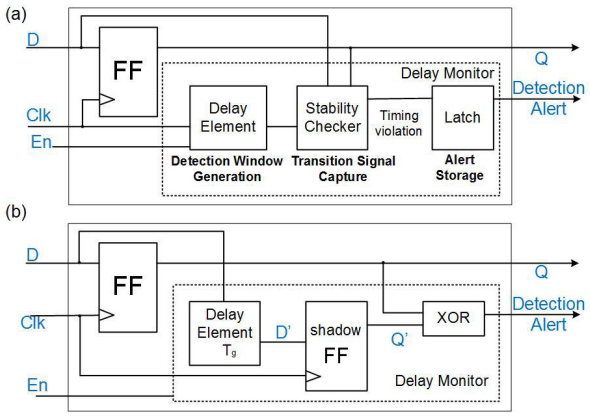

Fig. 1. Structure of a delay detecting flip-flop with the (a) stability checker [24, 27] or (b) comparator logic [24, 28]

To prevent the self-degradation of monitors, the enable signal *En* in Fig. 1 is used for periodic monitor activation for aging measurement [27].

B. Reuse of Delay Monitors for Testing

If a faulty transition occurs during the detection window, a delay fault is detected. With monitor reuse, some hidden delay faults violating the detection window are detectable at nominal speed, while they were before only detected by FAST. However, many hidden faults still remain unobservable. As shown in Fig. 2, assume pattern pair p_1 sensitizes the orange path *ABCE* to output O_1, and the last transition arrives at time t_1. When p_2 is provided, the green path *BDE* is sensitized and the output signal is stable after t_2. The transition of *Delayed Sig.* from *BDE* is undetectable due to the small fault size and short length of the propagation path. To further improve the fault coverage, the monitors can also be reused at FAST frequencies. If the test is repeated with lower clock period t_{FAST}, the detection window moves to the left of the time axis and the faulty transition is detected. At the same time, the transition caused by pattern pair p_1 at time t_1 in the fault-free circuit (in the following called *good transition*) violates the detection window as well.

To overcome this false positive detection, we can either remove the pattern pair that launches the culprit transition from the original test set or disable the corresponding monitor as explained in Section III. Signal *En* (Fig. 1 (b)) can be used to disable those monitors with false alerts.

III. Monitor and Pattern Selection for FAST

We aim to detect as many hidden delay faults as possible by reuse of aging monitors at the nominal and predefined FAST frequencies. The timing information of good and faulty transitions is collected by timing-accurate good machine and fault simulation and compared to the detection window of monitors at each test frequency. To this end, we use a high

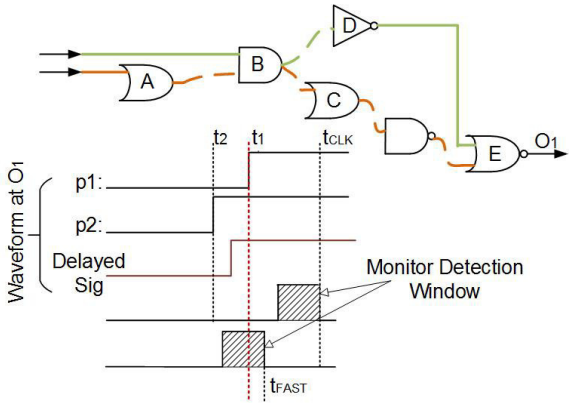

Fig. 2. The transition at t_1 in the fault-free circuit causes a false alert

throughput timing simulator optimized for GPUs [32]. A false alert is identified when a good transition occurs during the detection window ($tr^g = 1$, cf. Sec. III-A). If a monitor captures a faulty transition, a detection effect ($tr^f = 1$) is recorded for later optimization of the fault coverage. Usually the detection window size (T_g) of monitors is smaller than the slack of paths, thus good transitions trigger no false alert at nominal speed. To prevent false positive detections at monitors during FAST, the related pattern pairs are removed from the test set or the corresponding monitors are deactivated. We call the selected patterns and the assignments of monitor control signals a *test configuration*. A certain number of test configurations are generated at each frequency. For maximal coverage of HDFs at each frequency, a Pseudo-Boolean optimization is performed for pattern and monitor selection to generated each test configuration. Fault dropping is performed after each optimization step, and the subsequent test configuration is generated targeting the remaining undetected faults.

A. Modeling for Pseudo-Boolean Optimization

For every test configuration generation, a pseudo-Boolean optimization is performed. The objective function to be maximized is the number of detected target faults. The conditions for monitor false positive detections, i.e. no good transition lies in the detection window of monitors, are translated to Boolean constraints.

We model the pattern pairs in the pattern set $P = \{p_1, p_2, \ldots, p_Q\}$ with Boolean variables p_i. If $p_i = 1$, the pattern pair p_i is applied at the currently considered FAST frequency. If $p_i = 0$, the pair p_i is removed from the test set for that FAST frequency. Monitors inserted in the circuit are denoted by the set $M = \{m_1, m_2, \ldots, m_S\}$ of Boolean variables. Variable $m_j = 1$ if monitor m_j is enabled at the currently considered frequency.

In timing simulation, if pattern pair p_i causes the input of monitor m_j to toggle during the detection window, we model this transition with a Boolean variable tr and the equivalence $tr \Leftrightarrow p_i \wedge m_j$. If $tr = 1$, a monitor alert is generated at m_j when p_i is applied. If the transition is from the good machine simulation, we denote it as tr^g. When $tr^g = 1$, the good transition will trigger a false positive detection at the monitor. To prevent this, we provide the Boolean constraint $\bigvee_{l=1}^{L} tr_l^g = 0$, $l \in \{1, \ldots, L\}$, L is the number of transitions that trigger monitor false alerts, obtained from good machine simulation. Because such a good transition can be represented by the monitor and pattern combination above, we replace tr^g with $p_i \wedge m_j$ in the Boolean constraint and get $\bigvee_{l=1}^{L}(p_i \wedge m_j) = 0$. l encodes the indices (i, j). We negate the equation at both sides, and finally get the term that needs to evaluate to true:

$$\varphi_1 : \bigwedge\nolimits_{l=1}^{L}(\overline{p_i} \vee \overline{m_j}) \quad (i,j) \mapsto l \tag{1}$$

If the transition is the effect of a fault f in the target fault set F, we denote it as tr^f. When $tr^f = 1$, a faulty transition is detected by a monitor. A fault f is detected by observing at least one of its faulty transitions, which we model with help of a Boolean variable d_f: $d_f \Rightarrow \bigvee_{e=1}^{E_f} tr_e^f$, $e \in \{1, \ldots, E_f\}$. E_f is the number of faulty transitions resulting from that fault. These transitions are obtained from timing-accurate fault simulation. After transformation we obtain:

$$\varphi_2(f) : \overline{d_f} \vee (\bigvee\nolimits_{e=1}^{E_f} tr_e^f) \tag{2}$$

Using the Boolean equivalence $tr_e^f \Leftrightarrow p_i \wedge m_j$, we can express the condition when the effect of a fault tr_e^f is detected by a pattern and monitor. This equivalence can be transformed to $(\overline{tr_e^f} \vee (p_i \wedge m_j)) \wedge (tr_e^f \vee \overline{p_i \wedge m_j})$, and into conjunctive normal form (CNF) as:

$$\varphi_3(f, e) : (\overline{tr_e^f} \vee p_i) \wedge (\overline{tr^f} \vee m_j) \wedge (tr_e^f \vee \overline{p_i} \vee \overline{m_j}) \tag{3}$$

To maximize the fault coverage at each test frequency, we maximize the pseudo-Boolean function $\sum_{f \in F} d_f$. The Boolean constraints of Eq. 1 to Eq. 3 need to be satisfied for all faults:

$$\varphi_1 \wedge (\bigwedge\nolimits_{f \in F}(\varphi_2(f) \wedge \bigwedge\nolimits_{e=1}^{E_f} \varphi_3(f, e))).$$

To generate the test configuration, i.e. selected patterns and control assignment of monitors, for one frequency, this objective function and Boolean constraints are analyzed by a SAT solver. The Boolean assignments to P and M determine which pattern pairs are applied and which monitors are active.

B. Test Control Hardware

If the proposed monitor reuse approach is employed in a self-test or embedded deterministic test scenario, the patterns and test configuration information can be stored in the system in compressed form. The patterns are applied at the corresponding FAST frequencies according to the test configurations.

To set up the monitors in one test configuration, the on-chip control structure shown in Fig. 3 enables the selected monitors based on the test configuration index. This index is shifted in from the *Scan_in* signal. The decoding logic translates the index number into control assignments of monitors. The monitor configurations remains unchanged for all pattern pairs in the selected test configuration during testing.

To store N test configurations, $\lceil ld(N) \rceil$ scan flip-flops are required. The output of the OR gates is connected to the enable signal of the monitors m_i. During aging prediction, *Scan_en* is set to low and *Moni_en* is used to periodically

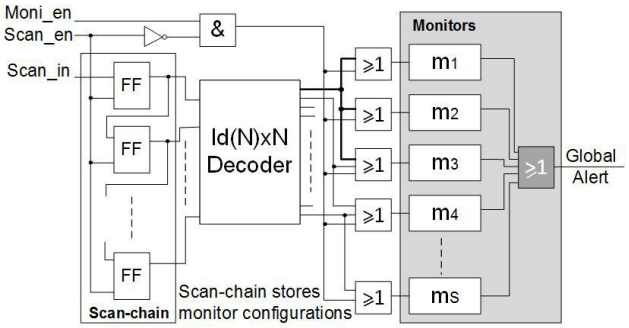

Fig. 3. Test control hardware for monitor selection

start the degradation measurement with all monitors. The OR gate (grey) collects *Detection Alert* (Fig. 1) of all monitors and generates a *Global Alert* if any monitor alert is issued.

When *Scan_en* is high, the circuit is switched into test mode. The values of the flip-flops control which test configuration is applied. In every test, one of the outputs of the decoder is high and at the same time all monitors connected to this end are enabled. For example (Fig. 3), when the first output (bold) of the decoder is high, monitors m_1, m_2, m_3 are activated.

The monitors and grey OR gate are integrated on-chip and reused. *Global Alert* indicates the test results. If required for diagnosis, the shadow registers (Fig. 1) can be included in a standard scan-chain. The delayed outputs (*Q'*) of shadow registers are read out only when the test case fails.

Thus, the hardware overhead is only induced by monitor activation control logic, which consists of $\lceil ld(N) \rceil$ scan flip-flops, S (S: number of monitors) OR gates, one inverter, one AND gate and a $\lceil ld(N) \rceil \times N$ decoder.

IV. Evaluation

A. Experiment Setup

In the experiments, we generate a compacted test pattern set targeting transition delay faults using a commercial tool. We assume monitors are placed at 25% of the pseudo outputs at long path ends [25]. When generating the test configurations for the monitor-based delay test, we target slow-to-rise and slow-to-fall small delay faults at inputs or outputs of all gates in the input cone of monitors. The delay fault size is set to 6σ, where σ is the standard deviation of the normal distribution of the process variation and set to 0.2 of the nominal gate delay. The nominal clock period is set to the critical path length plus a small time margin of 5%. Eight preselected FAST frequencies are evenly distributed from nominal (f) to three times of nominal frequency ($3f$). The size of detection window (T_g) is set to 50 ps.

The experiment is performed for ISCAS89, ITC99 and NXP benchmarks. The basic information of the circuits is listed in the first seven columns of Table I. The critical path length (*cpl*), number of patterns (*#patterns*) in the original test set and the number of monitors (*#monitors*) are listed in Col. 2 to 4 respectively. *#gates* is the number of gates in the combinational nets and *#flip-flops* is the number of flip-flops. The last three columns relate to the hardware overhead (cf. Sec. IV-D).

B. Generation of Test Configurations

Monitor and pattern selection allow to mask false alerts of monitors, but at same time reduce the observability of some detection effects. Some faults may only be activated by specific patterns and detectable at certain monitors. For a high coverage of target faults, two test configurations are generated for each of the eight FAST frequency to alleviate this effect. In this case, $N = 16$ test configurations are controlled by a 4 bit scan-chain and 4x16 decoder (Fig. 3). For each test configuration generation, a pseudo-Boolean optimization is performed.

The method is implemented in Java and uses the Sat4j library. It is executed on an intel Xeon core with 3.33 GHz.

The Pseudo-Boolean optimization is aborted when a timeout of 1 hour is reached. Fault dropping is done after each optimization, i.e. test configurations are generated for the remaining uncovered faults from the previous configurations. The runtime of the entire procedure is dominated by the exhaustive timing accurate fault simulation. The runtime of the Pseudo-Boolean optimization is constrained by the timeout.

The extension of the used timing simulator [33] also supports variation analysis during fault simulation, but it is beyond the scope of this paper.

C. Fault Site Coverage vs. Different Fault Size

To evaluate the effectiveness, fault simulations are performed for different fault sizes from 6σ to 30σ, with increments of 2σ. A fault at fault site si_x with size δ is denoted as $f^{\delta}_{si_x}$, where x is the fault site index. We say a fault site si^{δ}_x is *covered*, if at least one of the faults located at the fault site si_x with fault size smaller than or equal to δ is detected. For instance, $si^{10\sigma}_1$ is covered if any of the faults $f^{6\sigma}_{si_1}$, $f^{8\sigma}_{si_1}$ or $f^{10\sigma}_{si_1}$ is detected. The 16 test configurations generated for HDFs with size 6σ (cf. Sec. IV-B) are applied for all fault sizes (6σ to 30σ). The fault sites covered by the eight fixed frequencies w.r.t. a certain fault size δ are collected.

For two circuits, Fig. 4 depicts the the ratio ra^{δ}_{cov} of covered fault sites to all target fault sites ("cov" lines with the left vertical axis). ra^{δ}_{cov} increases when the fault size grows. Even if an HDF with small size is not detected initially, the monitors detect it if it grows in magnitude.

Fig. 4. The fault site coverage (left axis) and detection efficiency (right axis) by monitors w.r.t. different fault size

Multiple reasons constrain the coverage of fault sites. For instance, if the test pattern set does not sensitize a path through the fault site to a monitor with sufficiently small slack, the fault

TABLE I. Basic information of circuits

Benchmark *(1)*	cpl [ns] *(2)*	#patterns *(3)*	#monitors *(4)*	#gates *(5)*	#flip-flops *(6)*	A_{cut} [μm^2] *(7)*	A_{ctrl} [μm^2] *(8)*	A_{ctrl}/A_{cut} *(9)*
s9234	0.872	304	63	1766	228	4270	280	0.066
s13207	1.194	376	198	2867	669	10815	300	0.028
s15850	1.983	261	171	3324	597	10252	247	0.024
s35932	0.423	75	513	11168	1728	30781	722	0.023
s38417	1.186	251	436	9796	1636	28249	698	0.025
s38584	1.158	313	426	12213	1450	28250	593	0.021
b17	4.730	3855	379	26292	1415	38767	728	0.018
b18	6.070	5197	842	72359	3320	99818	1079	0.011
b19	5.440	7657	1679	143296	6642	199085	1877	0.009
p35k	3.192	2102	558	23294	2173	45456	569	0.013
p45k	2.306	5349	638	25406	2331	49567	580	0.012
p78k	1.701	136	872	70495	2977	94384	1424	0.015
p89k	2.886	1920	1140	58726	4301	99825	1051	0.011
p100k	2.759	5178	1458	60767	5735	119001	1441	0.012
p141k	2.828	1621	2626	107655	10501	213753	2290	0.011

site cannot be covered. In Fig. 4, we also show the detection efficiency of the monitor reuse approach for the different fault sizes. It is defined as *efficiency := #detected / #detectable*. It is shown by the "eff" lines with the secondary (right) vertical axis. For both circuits and all fault sizes, the efficiency is high and ranges from 0.87 to 0.96.

Table II tabulates the results for all circuits and investigates the reason for uncovered fault sites in more detail. Col. 2 shows the number of fault sites in each circuit. For a fault size of 30σ, the number of covered fault sites (#$cov^{30\sigma}$) is in Col. 3.

Col. 4 *#detectable* shows the upper limit of the number of fault sites that can possibly be detected by the monitor reuse approach. For this upper limit, we assume that each monitor can be selected individually in each cycle to reach the best observability of faults and at the same time prevent all false alerts of monitors. However, in practice, configuring monitors cycle per cycle is very costly. Thus, in our approach, monitors are controlled by test configurations that are applied per test frequency. The column *#detected* provides the number of covered fault sites with the 16 computed test configurations. As stated above, the efficiency of monitor reuse is the ratio *efficiency := #detected / #detectable* and listed in Col. 6. The efficiency ranges from 82.8 to 97.1%.

We analyzed in more detail the reasons that limits the detection of target fault sites and categorized the fault sites accordingly. The numbers of fault sites in each category are in Col. 7 to 10. We say a fault site is outside range (*#outside_range*) if the maximum fault size 30σ plus the length of the longest topological path through the fault site is smaller than the minimum test period $1/(3f)$, i.e. the faults at *outside range* sites can never be observed by a test frequency lower than or equal to $3f$. A fault site is unsensitized (*#unsensitized*) if no pattern in the test set activates a path from the fault site to a pseudo output with monitor. Some fault sites cannot be covered due to the preselected test frequencies. If no faulty transition through a certain fault site can trigger any monitor alert with all possible fault sizes at all frequencies, the fault site is sorted into frequency constraint and the number of them is listed in Col. *#freq_constraint*. The remaining fault sites (*#masking*) are uncovered because of the pattern and monitor selection for false alert masking. As shown in the results, the portion of fault sites uncovered due to pattern and monitor selection is quite low (1.4% - 8.8% of target fault sites). The *outside range* cause is the major reason for uncovered fault sites. For example for circuit p35k, 41.9% of the fault sites cannot be covered when the maximum FAST frequency is set to three times the nominal frequency. The principal limitations (Col. 7 to 10) bound the coverage ratio to 42.0% for p35k. In such cases, a higher coverage is possible if the test pattern set is extended by n-detect, timing-aware, or path-delay patterns.

D. Hardware Overhead

Col. 7 in Table I provides the area of circuit under test (CUT) A_{cut} in μm^2. The area is calculated as the sum of the combinational, sequential and monitor area. The absolute overhead of the monitor control logic A_{ctrl} and the relative overhead (ratio of the overhead to the original design A_{ctrl}/A_{cut}) are listed in the last two columns. The overhead ranges from 0.9% to 6.6% of the CUT and decreases for larger circuits.

V. Conclusion

In this work, we reuse in-situ aging monitors for efficient small delay fault tests. Thus the complex on-chip structure or expensive high-speed ATE for test response evaluation can be avoided. To mask the false alert of monitors at FAST, the relevant patterns of a test pattern set are selected, and the monitors are enabled or disabled according to the given test frequencies. Pattern and monitor selection are modeled as a Pseudo-Boolean optimization problem to maximize the fault coverage. Experimental results show that a high detection efficiency of small delay faults is achieved, ranging from 82.8% up to 97.1% with a very low hardware overhead.

Acknowledgements

This work was supported by the German Research Foundation (DFG) within the projects PARSIVAL (WU 245/16-1) and FAST (WU 245/19-1).

TABLE II. Target fault site coverage

Benchmark (1)	#fault_site (2)	#$cov^{30\sigma}$ (3)	#detectable (4)	#detected (5)	efficiency (6)	Uncovered Fault Sites #outside_range (7)	#unsensitized (8)	#freq_constraint (9)	#masking (10)
s9234	7178	6118	5869	5700	0.971	272	323	316	149
s13207	9348	6974	7118	6848	0.962	1448	537	119	270
s15850	10936	5499	5482	5278	0.963	4411	524	301	201
s35932	43390	40374	34918	30684	0.879	384	506	353	1773
s38417	36600	29260	28997	27323	0.942	3922	1148	647	1623
s38584	29630	23035	24229	22766	0.940	3896	695	559	1445
b17	82528	22812	21571	20090	0.931	39101	13934	5424	1257
b18	231041	74007	76145	72850	0.957	105022	34513	14208	3291
b19	441423	161900	164390	155329	0.945	185314	61183	23996	9030
p35k	102131	42926	42435	40580	0.956	42754	9740	4873	1838
p45k	85123	54623	57416	53804	0.937	20481	3172	3240	3607
p78k	327768	250458	256361	231262	0.902	26742	855	26343	23370
p89k	195872	68006	75874	64234	0.847	78867	22880	14499	11620
p100k	224464	122006	132245	120534	0.911	69324	12841	8677	11616
p141k	335571	148287	172655	142904	0.828	130000	15240	12570	29474

REFERENCES

[1] R. R. Montanes, J. P. de Gyvez, and P. Volf, "Resistance Characterization for Weak Open Defects," *IEEE Design Test of Computers*, vol. 19, no. 5, pp. 18–26, Sep 2002.

[2] R. Tayade, S. Sundereswaran, and J. Abraham, "Small-Delay Defect Detection in the Presence of Process Variations," in *Proc. Int'l Symp. on Quality Electronic Design (ISQED)*, March 2007, pp. 711–716.

[3] S. Natarajan, M. A. Breuer, and S. K. Gupta, "Process Variations and their Impact on Circuit Operation," in *Proc. IEEE Int'l Symp. on Defect and Fault Tolerance in VLSI Systems (DFTS)*, Nov 1998, pp. 73–81.

[4] M. Tehranipoor, K. Peng, and K. Chakrabarty, *Test and Diagnosis for Small-Delay Defects*. Springer, 2011.

[5] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Timing-Based Delay Test for Screening Small Delay Defects," in *43rd ACM/IEEE Design Automation Conference (DAC)*, 2006, pp. 320–325.

[6] N. Ahmed and M. Tehranipoor, "A Novel Faster-Than-at-Speed Transition-Delay Test Method Considering IR-Drop Effects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1573–1582, Oct 2009.

[7] X. Lin, K. h. Tsai *et al.*, "Timing-Aware ATPG for High Quality At-speed Testing of Small Delay Defects," in *15th Asian Test Symp. (ATS)*, Nov 2006, pp. 139–146.

[8] S. Hellebrand, T. Indlekofer *et al.*, "FAST-BIST: Faster-than-At-Speed BIST Targeting Hidden Delay Defects," in *Proc. IEEE Int'l Test Conf. (ITC)*, 2014, pp. 1–8.

[9] Y. M. Kim, Y. Kameda *et al.*, "Low-cost gate-oxide early-life failure detection in robust systems," in *Symp. on VLSI Circuits*, 2010, pp. 125–126.

[10] H. Yan and A. D. Singh, "On the Effectiveness of Detecting Small Delay Defects in the Slack Interval," in *Proc. of IEEE Int'l Workshop on Current and Defect Based Testing (DBT)*, April 2004, pp. 49–53.

[11] P. Nigh and A. Gattiker, "Test Method Evaluation Experiments and Data," in *Proc. IEEE Int'l Test Conf. (ITC)*, 2000, pp. 454–463.

[12] R. Vollertsen and R. Hijab, "Burn-in," in *IEEE Int'l Integrated Reliability Workshop Final Report*, 1999, pp. 132–133.

[13] W. W. Mao and M. D. Ciletti, "A Variable Observation Time Method for Testing Delay Faults," in *Proc. of 27th ACM/IEEE Design Automation Conf. (DAC)*, Jun 1990, pp. 728–731.

[14] H. Yan and A. D. Singh, "Experiments in Detecting Delay Faults using Multiple Higher Frequency Clocks and Results from Neighboring Die," in *Proc. IEEE Int'l Test Conf. (ITC)*, Sept 2003, pp. 105–111.

[15] S. Pei, H. Li, and X. Li, "An On-Chip Clock Generation Scheme for Faster-than-at-Speed Delay Testing," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2010, pp. 1353–1356.

[16] R. Tayade and J. A. Abraham, "On-chip Programmable Capture for Accurate Path Delay Test and Characterization," in *Proc. IEEE Int'l Test Conf. (ITC)*, Oct 2008, pp. 1–10.

[17] A. Singh, C. Han, and X. Qian, "An output compression scheme for handling X-states from over-clocked delay tests," in *Proc. 28th VLSI Test Symposium (VTS)*, April 2010, pp. 57–62.

[18] R. Carlsten, J. Ralston-Good, and D. Goodman, "An Approach to Detect Negative Bias Temperature Instability (NBTI) in Ultra-Deep Submicron Technologies," in *Proc. IEEE Int'l Symp. on Circuits and Systems (ISCAS)*, 2007, pp. 1257–1260.

[19] T.-H. Kim, R. Persaud, and C. Kim, "Silicon Odometer: An On-Chip Reliability Monitor for Measuring Frequency Degradation of Digital Circuits," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 874–880, 2008.

[20] A. Ghosh, R. Brown *et al.*, "A Precise Negative Bias Temperature Instability Sensor using Slew-Rate Monitor Circuitry," in *Proc. IEEE Int'l Symp. on Circuits and Systems (ISCAS)*, 2009, pp. 381–384.

[21] A. Drake, R. Senger *et al.*, "A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor," in *Proc. IEEE Int'l Solid-State Circuits Conf.*, Feb 2007, pp. 398–399.

[22] S. Wang, J. Chen, and M. Tehranipoor, "Representative Critical Reliability Paths for Low-Cost and Accurate On-Chip Aging Evaluation," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design (ICCAD)*, Nov 2012, pp. 736–741.

[23] S. Das, C. Tokunaga *et al.*, "RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan 2009.

[24] M. Agarwal, B. C. Paul *et al.*, "Circuit Failure Prediction and Its Application to Transistor Aging," in *Proc. 25th IEEE VLSI Test Symposium (VTS)*, May 2007, pp. 277–286.

[25] M. Agarwal, V. Balakrishnan *et al.*, "Optimized Circuit Failure Prediction for Aging: Practicality and Promise," in *Proc. IEEE Int'l Test Conf. (ITC)*, Oct. 2008, pp. 1–10.

[26] H. Dadgour and K. Banerjee, "Aging-resilient design of pipelined architectures using novel detection and correction circuits," in *Proc. Design, Autom. and Test in Europe Conf. (DATE)*, 2010, pp. 244–249.

[27] J. Vazquez, V. Champac *et al.*, "Programmable aging sensor for automotive safety-critical applications," in *Proc. Design, Automation Test in Europe Conference (DATE)*, march 2010, pp. 618–621.

[28] M. Saliva, F. Cacho *et al.*, "Digital circuits reliability with in-situ monitors in 28nm fully depleted SOI," in *Proc. Design, Automation Test in Europe Conf. Exhibition (DATE)*, March 2015, pp. 441–446.

[29] W. Wang, Z. Wei *et al.*, "An Efficient Method to Identify Critical Gates under Circuit Aging," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design (ICCAD)*, Nov. 2007, pp. 735–740.

[30] L. Lai, V. Chandra *et al.*, "Slackprobe: A flexible and efficient in situ timing slack monitoring methodology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 8, pp. 1168–1179, Aug 2014.

[31] C. Liu, M. A. Kochte, and H. J. Wunderlich, "Efficient observation point selection for aging monitoring," in *Proc. IEEE 21st International On-Line Testing Symposium (IOLTS)*, July 2015, pp. 176–181.

[32] E. Schneider, S. Holst *et al.*, "Gpu-accelerated small delay fault simulation," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2015, pp. 1174–1179.

[33] E. Schneider, M. A. Kochte *et al.*, "GPU-Accelerated Simulation of Small Delay Faults," *in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2016.