Using Programmable Delay Monitors for Wear-Out and Early Life Failure Prediction

Liu, Chang; Schneider, Eric; Wunderlich, Hans-Joachim

Proceedings of the ACM/IEEEConference on Design, Automation Test in Europe (DATE'20), Grenoble, France, 2020, pp. 804–809

doi: https://doi.org/10.23919/DATE48585.2020.9116284

Abstract: Early life failures in marginal devices are a severe reliability threat in current nano-scaled CMOS devices. While small delay faults are an effective indicator of marginalities, their detection requires special efforts in testing by so-called Faster-than-At-Speed Test (FAST). In a similar way, delay degradation is an indicator that a device reaches the wear-out phase due to aging. Programmable delay monitors provide the possibility to detect gradual performance changes in a system and allow to observe device degradation.This paper presents a unified approach to test small delay faults related to wear-out and early-life failures by reuse of existing programmable delay monitors within FAST. The approach is complemented by a test-scheduling which optimally selects frequencies and delay configurations to significantly increase the fault coverage of small delays and to reduce the test time.

Preprint

Using Programmable Delay Monitors for Wear-Out and Early Life Failure Prediction

Chang Liu, Eric Schneider and Hans-Joachim Wunderlich
University of Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Germany
{chang.liu,schneiec}@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de

***Abstract*—Early life failures in marginal devices are a severe reliability threat in current nano-scaled CMOS devices. While small delay faults are an effective indicator of marginalities, their detection requires special efforts in testing by so-called Faster-than-At-Speed Test (FAST). In a similar way, delay degradation is an indicator that a device reaches the wear-out phase due to aging. Programmable delay monitors provide the possibility to detect gradual performance changes in a system and allow to observe device degradation.**

This paper presents a unified approach to test small delay faults related to wear-out and early-life failures by reuse of existing programmable delay monitors within FAST. The approach is complemented by a test-scheduling which optimally selects frequencies and delay configurations to significantly increase the fault coverage of small delays and to reduce the test time.

***Keywords*-Faster-than-at-speed test, small delay faults, aging monitors, programmable delay monitors**

I. Introduction

Technology scaling causes devices to become more susceptible to aging mechanisms such as *Bias Temperature Instability* (BTI), *Hot-Carrier Injection* (HCI), or *Electro Migration* (EM), which shift parameters in transistors and interconnects and gradually degrade the circuit performance over the lifetime [1]. At the same time, young devices can also fail in their *early life* phase after deployment due to marginalities, even though they have passed the manufacturing and burn-in tests prior to shipment [2].

To prevent fatal consequences in safety-critical systems and applications, persistent performance monitoring and accurate failure prediction is required. Aging monitors [3, 4] are a common technique for periodic measurements of the device performance and allow to issue aging alerts upon reaching a certain level of degradation. To enable early countermeasures to mitigate aging effects and to prolong degradation, the integration of programmable delay monitors [5, 6] is demanded.

Device degradation due to aging as well as early life failures caused by marginal devices are often indicated by the presence of small delay faults in the circuit [2]. The delay introduced by these faults can magnify quickly after a short term of operation and eventually cause a performance degradation until the device begins to fail. However, the fault size is typically too small to be detected with at-speed test approaches even with timing-aware test patterns. Therefore, this type of small delay fault is also referred to as *"hidden delay fault"* (HDF). Again, although such HDFs do not influence the circuit functionality at the very moment of manufacturing, they may impose a severe reliability risk after deployment. Therefore it is crucial to test for these faults to identify early-life and aging related problems as early as possible.

An effective method for detecting hidden delay faults is *Faster-than-At-Speed-Test* (FAST), which applies tests at frequencies higher than the nominal operational speed [7, 8]. However, the *maximum FAST frequency* f_{max} is often limited [9–11] by physical properties of the design (e.g., parasitic capacitance or IR-drop [12]). This restriction of f_{max} considerably restricts the efficacy for hidden delay fault detection.

In [13] a unified capture circuit was designed to generate clock signals of FAST and detection windows for signal stability checking as required for aging prediction. While the goal of the work was the reduction of the hardware penalty, it did not target the improvement of the fault coverage or monitoring accuracy.

Since delay monitors can sense the delay deviation in circuit paths, [14] proposed to reuse them for small delay fault testing. The method directly applied aging monitors for delay test evaluation, evading extra infrastructures, e.g., an ATE, MISR or X-tolerant compactors [15, 16]. However, the reachable fault coverage is constrained by unoptimized fixed test frequencies and maximum test speed. [17] extended delay monitors at a low cost to achieve data signal sampling at a halved FAST clock period. As a result, the small delay fault coverage can be significantly improved and also allows to minimize the number of test frequencies and test time. However, even with the novel monitor structure, some of the fault effects remain not observable due to the constraint of the maximum test frequency.

In this work, the delay elements in programmable aging monitors are utilized to shift fault effects into the observable FAST frequency range. This way, previously undetected HDFs can be propagated to pseudo-outputs with monitors for detection at lower test frequencies or even at-speed. In addition, the programmable feature of the monitors provides more flexibility for test schedule optimization, such that all target hidden delay faults can be covered with the minimal test time during FAST. The overall contributions of this work are:

- Utilization of programmable delay monitors to increase HDF coverage and reduce test frequencies in FAST.
- A test scheduling based on Integer Linear Programming (ILP), to produce an optimal frequency-, test pattern- and monitor configuration selection with minimal test time.

The remainder of the paper is organized as follows: Section II gives background on small delay fault testing and pro-

grammable aging monitors. Section III provides an overview of applying programmable aging monitors for small delay fault detection and explains the workflow of the test procedure. In Section IV, an optimal test scheduling procedure is presented to obtain the required target fault coverage with minimal test time. Finally, the presented method is evaluated in Section V.

II. Background

A. Detection Range of Small- and Hidden Delay Faults

A *small (gate) delay fault* $\varphi := (g, \delta)$ is considered a lumped manifestation at a fault site g (i.e., gate pin) in a circuit that increases the propagation delay of a signal transition through g by a certain amount of time $\delta \in \mathbb{R}$ [18]. The *fault size* δ (i.e., delay introduced by the fault) is typically smaller than the clock period and the behavior of φ is usually not reflected by simpler fault models, such as *transition faults* [19].

Definition 1: If a small delay fault φ is *not* detected by a test set P for a given test frequency f, then φ is considered as *hidden delay fault* (HDF) with respect to P and f. For a set of frequencies F, the set of all hidden delay faults with respect to P and F is denoted as $\Phi_{HDF}(P, F)$.

In FAST, all observation times t at circuit outputs are selected from the interval (t_{min}, t_{nom}) whose boundaries are defined by the maximum (f_{max}) and nominal frequency (f_{nom}) through $t_{min} := 1/f_{max}$ and $t_{nom} := 1/f_{nom}$.

Definition 2: Let $t \in (t_{min}, t_{nom})$ be an observation time. For a small delay fault φ and test pattern set P, time t is called a *detecting observation time* (DOT), if there is a pattern $p \in P$, such that φ is detected by capturing the responses at time t. The set of *all* DOTs is called the *detection range* $I(\varphi, P)$ *of* φ with respect to P. Usually, $I(\varphi, P)$ is not a contiguous range, but a union of intervals.

This work follows a pessimistic approach to reflect pulse filtering in CMOS technology. If the length of an interval in the detection range is below a specified threshold, then the interval is assumed to be a *glitch* and is not added to the detection range of a fault. In the example of Fig. 1, the small glitch between the intervals I_1 and I_2 is shorter than the required threshold and hence is not included in $I(\varphi, P)$. If a glitch masks a fault then the adjacent surrounding intervals (e.g., between I_2 and I_3) are kept pessimistically as disjoint intervals. Also, all detection intervals outside of t_{min} and t_{max} are ignored.

B. Programmable Delay Monitor

A commonly used type of in-situ aging monitors is a delay detecting flip-flop which is a standard flip-flop extended with a delay monitor [3]. The delay monitors check the signal stability during a predefined detection window (i.e., *guard band*). Aging alerts are issued for indicating imminent timing failures if the observed signal toggles during the detection window. A delay monitor often consists of delay elements, a shadow flip-flop, and an XOR gate. Fig. 2 (a) shows a programmable delay monitor with four different delay elements. The delay element is selected by the control signal *Sel* of the MUX [6].

At the beginning of the circuit lifetime, a large delay element is selected to sense the initial degradation state of the circuit (Fig. 2 (b)). To detect aging- or stress-related

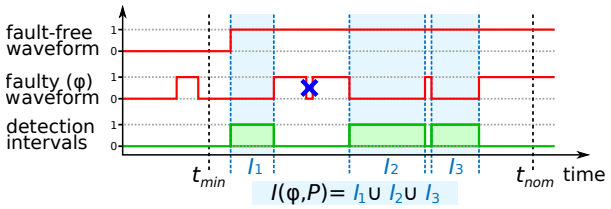

Fig. 1. *Detection range* $I(\varphi, P)$ of a fault φ for a pattern set P computed from *detection intervals* I_1, I_2 and I_3 after pulse filtering [17].

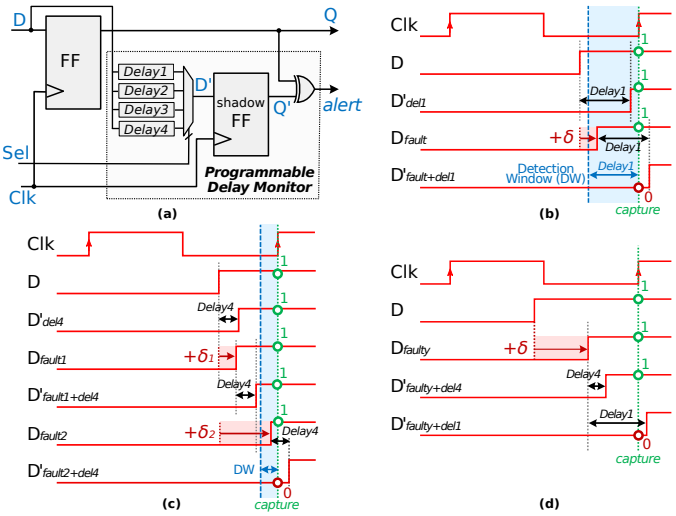

Fig. 2. Programmable delay monitor [6]: a) Structure, b) aging prediction with large delay element configuration, c) small delay element configuration and d) hidden delay fault detection.

delay deviations, a large delay (i.e., *Delay1*) is chosen to generate a wide detection window (highlighted region). When the monitored signal D is stable during the detection window, D'_{del1} is also stable before the rising edge of *Clk*. In this case, the identical logic value is captured by both standard and shadow registers and no aging alert is generated. If the observed signal is degraded by δ_1 and D_{fault} toggles within the detection window, the shadow register captures a logic '0' from $D'_{fault+del1}$, which varies from the standard flip-flop value ('1'). The captured states Q and Q' are compared by the XOR gate and an *alert* is issued.

After the first alert, aging countermeasures such as frequency or voltage scaling can be enabled to reduce further degradation. To measure the aging process within a degraded circuit, the programmable monitor should select a smaller delay (i.e., *Delay4* in Fig. 2 (c)). The slightly delayed signal D_{fault1} is now stable during the narrowed detection window with sufficient slack such that no alert is caused. After continued degradation, the latest transition of the further degraded signal D_{fault2} will violate the narrow detection window and trigger an alert again thus indicating an imminent failure.

III. Programmable Delay Monitor for Small Delay Fault Detection

Fig. 3 compares the hidden delay fault coverage of FAST in an industrial design *with* and *without* the use of output monitors, which was obtained from fault simulation of a

predefined test set. A fault size of $\delta = 6\sigma$ was chosen in order to model degraded or marginal devices, where σ is the standard deviation of process variation with a value of 20% of the nominal gate delay.

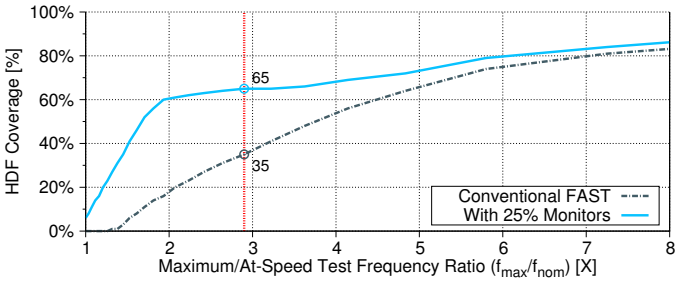

Fig. 3. Hidden delay fault (HDF) coverage for different maximum test frequencies f_{max} during FAST.

The dashed line represents the maximum HDF coverage reachable by conventional FAST with test frequencies in the range from nominal f_{nom} to the *maximum* FAST frequency f_{max}. The higher f_{max}, the higher the achieved fault coverage, since the detection of the fault effects requires high test speeds to overcome the large slack, especially when propagated along short paths only. As shown, about 35% of HDFs can be detected by the conventional FAST for the maximum test frequency $f_{max} \approx 2.9 \cdot f_{nom}$. Typically, f_{max} is bounded by $3 \cdot f_{nom}$ [9–11].

If programmable delay monitors are integrated on-chip for aging prediction, then not only the standard flip-flops in the circuit but also the shadow registers in monitors can be used for HDF detection. As shown in Fig. 2 (a), signal D' observed by the shadow register is the delayed data signal D. Because of the delay element, the fault effects of HDFs in D are shifted to a lower frequency range. For instance, the fault in Fig. 2 (d) can be detected by the shadow flip-flop at nominal test period when a large delay (*Delay1*) is configured, although it was unobservable without the help of monitors or with a smaller delay (*Delay4*) configuration. The solid curve in Fig. 3 represents the HDF coverage, when monitors are inserted in 25% of pseudo-outputs at long path ends with the delay of $\frac{1}{3} \cdot t_{nom}$. As shown, a noticeable increase in HDF coverage can be achieved for lower test frequencies now reaching up to 65% for the frequency range $[f_{nom}, 3 \cdot f_{nom}]$.

A. Overview of the Test Flow with Monitors

Fig. 4 provides an overview of the implemented HDF test flow. First, a topological analysis of the circuit is performed (1) using timing information from *standard delay format* files. During the process, all *at-speed detectable* faults (with minimum slack smaller than the fault size) and *timing-redundant* HDFs (observable only by pseudo-outputs without monitor integration and demanding test frequencies higher than f_{max}) are identified and removed from the initial fault list. Explicit timing-accurate fault simulation [20] is used to investigate the remaining fault sites (2). By comparing the fault-free and faulty waveforms in simulation, the *detection ranges* (DR) of each fault are determined (3). Based on the information about the monitor configurations, the detection ranges of HDFs are analyzed with respect to the different delay elements (4) under which some previously undetected faults can become now observable at-speed (cf. Sec. III-B). After removal of these new at-speed monitor detectable faults, the remaining faults constitute the set of *target faults* (5) which can only be detected by using FAST frequencies. The detection ranges of the target faults are used to determine the relevant observation times and hence the candidate space of test frequencies $F \subseteq [f_{nom}, f_{max}]$ for detection. Finally, an optimization method is proposed to generate an efficient *test schedule* for covering all the target HDFs with minimal test time (6). A test schedule $S \subseteq F \times P \times C$ is a set of frequency-, pattern- and delay-configuration combinations, where F is the set of available test frequencies, P is the original test set and C is the set of all monitor delay configurations. Each combination $(f, p, c) \in S$ indicates the required test frequency $f \in F$ for the application of a test pattern pair $p \in P$ under a given monitor configuration $c \in C$.

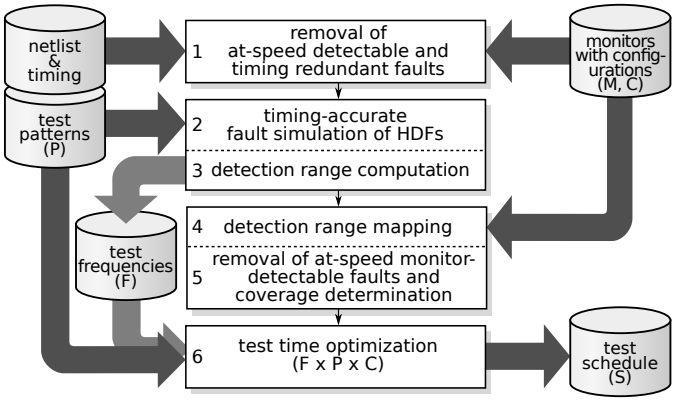

Fig. 4. Overview of the proposed HDF test flow.

B. Detection Range Shifting

In general, the detection range of a fault at a circuit output is computed by XOR-ing the fault-free and faulty output waveforms. The union of all XORed output waveforms of all output signals influenced by the fault φ then represents the detection range $I(\varphi)$ of the fault. As both standard flip-flops (FF) and shadow registers (SR) in aging monitors contribute to the observability of small delay faults, their respective detection ranges $I_{FF}(\varphi)$ and $I_{SR}(\varphi)$ are determined separately for each fault φ, such that

$$I(\varphi) := I_{FF}(\varphi) \cup I_{SR}(\varphi).$$

With $f_{max} = 3 \cdot f_{nom}$, the detection ranges of standard flip-flops $I_{FF}(\varphi) \subseteq [t_{min}, t_{nom}] = [\frac{1}{3} \cdot t_{nom}, t_{nom}]$, i.e. the fault effects of φ outside the frequency range are unobservable. Delay elements of monitors shift the observed signal (and the fault effects) to a lower frequency range, i.e., the detection range of a shadow register at an output o is the detection range of its standard flip-flop shifted d time units to the right along the time axis, where d is the selected delay of the monitor. This delivers

$$I_{SR}(\varphi, o) := I_{FF}(\varphi, o) + d.$$

Hence, the shadow registers allow to modify the detection range of faults in the following two ways:

- Previously *unobservable* fault effects, i.e., detection ranges located in $(0, \frac{1}{3} \cdot t_{nom})$ can be shifted to *testable* ranges by a maximum monitor delay of $d := \frac{1}{3} \cdot t_{nom}$.
- The delay element settings can shift the detection range of faults for detection by different test frequencies: $I_{SR}(\varphi, o) = \cup_{d \in C}[I_{FF}(\varphi, o) + d]$, where C is the set of configurable delays in a monitor.

Thus, delay configurations allow to increase the detection ranges, which not only improves the HDF coverage but also provides possibilities for test time optimization in FAST.

IV. Test Schedule Optimization

A. Test Observation Time Discretization

The detection range of a fault is used to determine all relevant FAST clock periods for detection. To reduce the search space and computational effort of test frequency selection, candidates of clock periods are generated by discretizing the continuous detection intervals of faults. The discretization procedure is demonstrated in Fig. 5, which shows the detection ranges of three faults φ_1, φ_2 and φ_3. The boundaries of the detection intervals divide the time axis into six intervals. The observation times of an interval all detect identical faults (number shown above). Time intervals in which more faults are detected are considered as *representative intervals*. In order to cover the targeted faults robustly even under variations, the mid-points of the representative intervals are selected (e.g., T_0 and T_1) as candidates for test clock period/frequency selection.

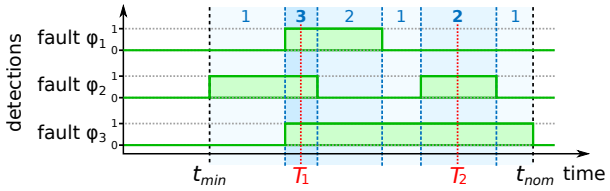

Fig. 5. Example of test observation time discretization for observation time selection. The number of detected faults in each interval is shown on the top.

B. Problem Statement

Switching of frequencies during FAST usually requires a re-locking of the Phase Locked Loop (PLL) structure in clock generators, which takes tens or hundreds of microseconds, corresponding to a loss of several thousands of instruction cycles [21, 22]. Consequently, the number of frequencies needed for FAST often has a larger impact on test time than the number of patterns and monitor configurations. As a result, the test time optimization process is split into two steps:

1) aim to choose a minimum number of test frequencies
2) for each selected test frequency, optimize the pattern set and the monitor configuration.

Each step of the optimization is a set-covering problem. Assume a set of test clock period candidates $T := \{t_1, t_2, ..., t_L\}$. Let $\Phi_i \subset \Phi^{tar}$ be the set of faults detected at time t_i. The union of detected faults at all possible capture times is $\cup_{i=1}^{L} \Phi_i = \Phi^{tar}$. The first optimization step is to find a collection $\Psi \subseteq \{\Phi_1, \Phi_2, ..., \Phi_L\}$ that satisfies $\cup_{\psi \in \Psi} \psi = \Phi^{tar}$ and has minimum $|\Psi|$.

After frequency optimization, the optimized test clock period set T^{opt} is obtained. Let Φ_j^{opt} be the set of faults which is covered at clock period $t_j^{opt} \in T^{opt}$. It is assumed that all monitors share the identical delay setting and each monitor configuration can be applied concurrently during shift-in of the test patterns. To minimize the test time at each chosen test-speed, let $\Phi_{(m,n)}$ be the set of faults activated by the m-th pattern in the original test set, which are detected under the n-th monitor configuration, with $m \in \{1, 2, ..., |P|\}$ and $n \in \{1, 2, ..., |C|\}$. In the second optimization step, a set of collections $\{\Omega_1, \Omega_2, ..., \Omega_{|T^{opt}|}\}$ is determined. Each Ω_j corresponding to $t_j \in T^{opt}$ is a collection of subsets $\Phi_k \in \{\Phi_1, \Phi_2, ..., \Phi_K\}$ with $(m, n) \rightarrow k \in \{1, 2, ..., K\}$ that satisfies $\cup_{\omega \in \Omega_j} \omega = \Phi_j^{opt}$ with minimum $|\Omega_j|$.

C. Integer Linear Programming

Similar problems with NP-hard complexity have been solved by a hypergraph-based method in [23] and a heuristic selection algorithm in [17] to reduce the computational effort. For an optimal solution, the work at hand models the set-covering problems as zero-one linear programming that is solved by a commercial tool. In the first optimization step (test frequency selection), a set of Boolean variables $T^B := \{t_1^B, t_2^B, ..., t_{|T|}^B\}$ is defined. Each variable $t_i^B \in T^B$ equals '1', when the corresponding test clock period candidate $t \in T$ is selected for FAST. To minimize the number of selected test clock periods, the objective function of the zero-one linear programming is set as:

$$\text{minimize}\Big(\sum_{t^B \in T^B} t^B\Big)$$

subject to:

$$\sum_{t_i \in T^\varphi} t_i^B \geq 1, \forall \varphi \in \Phi^{tar}$$

Here, T^φ is the set of test clock periods that detect fault φ. For a given target fault set, the expression above describes that each fault is detected at least once in any of its detection intervals.

For each selected test clock $t_j^{opt} \in T^{opt}$, the fault set Φ_j^{opt} *required* to be covered is identified by sorting the test clock periods T^{opt} with a heuristic selection that uses fault dropping. In the second optimization step, relevant test patterns and monitor configurations are chosen accordingly. For this, corresponding variables $p_m c_n$ are defined, which evaluate to '1', if a fault is observed by the m-th pattern under the n-th monitor configuration. To minimize the test time of each selected frequency, the number of pattern-configuration combinations needs to be minimized. The objective function is written as:

$$\text{minimize}\Big(\sum_{(m,n) \in P_{idx} \times C_{idx}} p_m c_n\Big),$$

where P_{idx} is the set of pattern indices in the original test set and C_{idx} is the set of monitor configuration indices. The minimization of the objective function is subject to:

$$\sum_{p_m c_n \in PC^\varphi} p_m c_n \geq 1, \forall \varphi \in \Phi_j^{opt},$$

with PC^{φ} being the set of pattern-configuration combinations that detect fault φ. Thus, fault φ needs to be observed at least once by the pattern-configuration combination $p_m c_n \in PC^{\varphi}$. This constraint needs to hold for all faults covered at the test clock period t_j^{opt}.

V. Evaluation

For the experiments, benchmark circuits from ISCAS'89 and industrial designs were synthesized using the NanGate 45nm open cell library [24]. The nominal clock period (clk) of each circuit is set as the critical path length (cpl) from static timing analysis plus 5% margin (i.e., $clk := 1.05 \cdot cpl$). During synthesis, monitors are integrated at long path ends [25] covering 25% of the total pseudo-primary outputs. Each monitor has four different delay elements with delays $d := 0.05 \cdot clk$, $0.1 \cdot clk$, $0.15 \cdot clk$ and $\frac{1}{3} \cdot clk$. In this work it is assumed that all monitors share the identical delay setting for any given configuration. Compacted transition delay fault test sets with an average test coverage of over 99.9% are used for the evaluation, which were generated by a commercial ATPG tool. As initial fault set, small delay faults with a fault size of $\delta = 6\sigma$ (cf. Sec. III) are considered at all input and output pins of gates in the circuit. Furthermore, two individual small delay faults are modeled at each location to distinguish *slow-to-rise* and *slow-to-fall* effects.

All experiments were performed on a host system equipped with two Intel Xeon E5-2687 v2 processors clocked at 3.4GHz with access to 256GB of main memory. For the timing-accurate small delay fault simulation, the fast parallel GPU-based simulator of [20] was used, which was executed on an NVIDIA Tesla P100 accelerator. The optimization of the ILP solver was aborted when a timeout of 1 hour was reached, which happened only for a few circuits.

A. Increased Hidden Delay Fault Coverage

Table I summarizes the basic circuit statistics and the number of faults detected by conventional FAST and the novel monitor-reuse method (cf. Sec. IV). Column 2 and 3 report the size of the circuit in the number of gates along with the number of flip-flops. The size of the ATPG-generated test pattern set is given in column 4. In column 5, the number of monitors at path ends ($|M|$) is shown. Column 6 through 8 summarize the number of faults detected by conventional FAST without (*conv.*) and the presented approach with programmable monitors (*prop.*) as well as the relative gain in HDF coverage. Note that the delay configurations of the monitors allow for detection of additional HDFs under nominal frequency. These at-speed monitor-detectable faults were thus removed from the target fault set Φ_{tar}. The remaining number of faults in the target set is listed in the last column.

B. Test Schedule Optimization Results

Table II summarizes the required test frequencies, test pattern pairs and monitor configurations in the test schedule resulting from the ILP selection algorithm (cf. Sec. IV-C) to achieve full coverage of targeted HDFs (cf. Table I).

Column 2 through 5 list the results of test frequency selection. Column 2 and 3 compare the number of required test frequencies $|F|$ with conventional FAST (*conv.*) and with a heuristic solution (*heur.*) presented in [17]. Column 4 shows the number of frequencies selected by the presented approach (*prop.*) with the use of programmable monitors. The relative reduction $\Delta\%_{|F|} := (1 - |F_{prop.}|/|F_{conv.}|) \cdot 100\%$ is given in column 5. The last three columns state the test time before (*orig.*) and after (*opti.*) the optimal test schedule generation, as well as the relative time reduction $\Delta\%_{|PC|} := (1 - |S|/|P \times C \times F|) \cdot 100\%$.

The number of test frequencies selected by the ILP algorithm (column 4) is always smaller than the heuristic method (column 3), pointing out the efficiency of the ILP approach. As shown in column 5, the relative reductions in the number of frequencies are positive for most of the benchmarks, indicating that with the help of programmable delay monitors more hidden delay faults can be detected with less testing time. While for circuit p45k, the number of test frequencies is reported to be 8.3% higher, the relative gain in HDF detection (column 8 in Table I) shows an significant increase in the HDF coverage by 65.2% with the programmable delay monitors. Yet, a noticeable reduction in test time can be observed in the last column of Table II ranging from 73.1% to 98.4%.

Table III summarizes the required test frequencies F_{cov}, the total combinations of test pattern-configurations without scheduling PC_{cov} and the optimal test schedule S_{cov} obtained by the presented method with respect to a given coverage cov of targeted HDFs (cf. Table I). For each required coverage, the relative test time reduction ($\Delta\%$) is calculated as $(1 - |S_{cov}|/|PC_{cov}|) \cdot 100\%$ The respective results of each coverage target are given in consecutive columns.

As shown, in all cases significant test time reduction was achieved. Compared to $cov = 100\%$ (cf. Table II), the number

TABLE I. Circuit statistics and targeted hidden delay faults (HDF).

Circuit[1]	Gates[2]	FFs[3]	$\lvert P\rvert$[4]	$\lvert M\rvert$[5]	Detected HDFs			Φ_{tar}[9]
					conv.[6]	*prop.*[7]	$\Delta\%$[8]	
s9234	1766	228	155	63	5469	6135	(+12.2%)	4655
s13207	2867	669	195	198	3349	7859	(+134.7%)	6814
s15850	3324	597	134	169	3541	8880	(+150.8%)	8607
s35932	11168	1728	39	513	34868	36129	(+3.6%)	16211
s38417	9796	1636	128	435	25064	32014	(+27.7%)	26327
s38584	12213	1450	160	426	20348	31119	(+52.9%)	29608
p35k	23294	2173	1518	558	35669	59759	(+67.5%)	53592
p45k	25406	2331	2719	638	48764	80544	(+65.2%)	79752
p78k	70495	2977	70	872	325682	337977	(+3.8%)	245824
p89k	58726	4301	993	1140	45792	133175	(+190.8%)	132503
p100k	60767	5735	2631	1458	111955	206990	(+84.9%)	197007
p141k	107655	10501	824	2626	196491	297260	(+51.3%)	290637

TABLE II. Number of selected test frequencies and test time in comparison.

Circuit[1]	selected frequencies $\lvert F\rvert$				pattern-configs. $\lvert P{\times}C\rvert$		
	conv.[2]	*heur.*[3]	*prop.*[4]	$\Delta\%_{\lvert F\rvert}$[5]	*orig.*[6]	*opti.*[7]	$\Delta\%_{\lvert PC\rvert}$[8]
s9234	20	16	13	35.0%	10075	662	(+93.4%)
s13207	17	16	12	29.4%	11700	852	(+92.7%)
s15850	24	25	22	8.3%	14740	949	(+93.6%)
s35932	16	8	7	56.3%	1365	367	(+73.1%)
s38417	34	23	18	47.1%	11520	1954	(+83.0%)
s38584	31	23	17	45.2%	13600	1823	(+86.6%)
p35k	58	49	40	31.0%	303600	6857	(+97.7%)
p45k	24	36	26	-8.3%	353470	5576	(+98.4%)
p78k	47	34	29	38.3%	10150	2323	(+77.1%)
p89k	44	52	41	6.8%	203565	10790	(+94.7%)
p100k	46	51	40	13.0%	526200	13577	(+97.4%)
p141k	60	65	48	20.0%	197760	17762	(+91.0%)

TABLE III. Test time reduction: Number of required test frequencies $|F_{cov}|$ for different coverages cov of targeted HDFs. Total number pattern-configuration combinations $|PC_{cov}|$ from naïve approach and test schedules S_{cov} generated by the presented method for the selected test frequencies F_{cov}.

Circuit[1]	Targeted Hidden Delay Fault Coverage (cov)															
	$cov \geq 99\%$				$cov \geq 98\%$				$cov \geq 95\%$				$cov \geq 90\%$			
	$\|F_{99}\|$[2]	$\|PC_{99}\|$[3]	$\|S_{99}\|$[4]	$\Delta\%$[5]	$\|F_{98}\|$[6]	$\|PC_{98}\|$[7]	$\|S_{98}\|$[8]	$\Delta\%$[9]	$\|F_{95}\|$[10]	$\|PC_{95}\|$[11]	$\|S_{95}\|$[12]	$\Delta\%$[13]	$\|F_{90}\|$[14]	$\|PC_{90}\|$[15]	$\|S_{90}\|$[16]	$\Delta\%$[17]
s9234	9	6975	640	(+90.8)	8	6200	629	(+89.9)	5	3875	558	(+85.6)	4	3100	528	(+83.0)
s13207	9	8775	831	(+90.5)	7	6825	799	(+88.3)	5	4875	752	(+84.6)	4	3900	695	(+82.2)
s15850	13	8710	896	(+89.7)	10	6700	851	(+87.3)	7	4690	771	(+83.6)	5	3350	674	(+79.9)
s35932	6	1170	357	(+69.5)	5	975	341	(+65.0)	4	780	318	(+59.2)	3	585	285	(+51.3)
s38417	10	6400	1836	(+71.3)	8	5120	1723	(+66.4)	6	3840	1557	(+59.5)	4	2560	1294	(+49.5)
s38584	9	7200	1678	(+76.7)	7	5600	1606	(+71.3)	5	4000	1454	(+63.7)	3	2400	1172	(+51.2)
p35k	22	166980	6569	(+96.1)	17	129030	6351	(+95.1)	10	75900	5659	(+92.6)	7	53130	5177	(+90.3)
p45k	10	135950	5232	(+96.2)	7	95165	4987	(+94.8)	4	54380	4396	(+91.9)	2	27190	3573	(+86.9)
p78k	6	2100	1443	(+31.3)	5	1750	1291	(+26.2)	3	1050	949	(+9.6)	2	700	654	(+6.6)
p89k	20	99300	10140	(+89.8)	15	74475	9697	(+87.0)	10	49650	8723	(+82.4)	6	29790	7261	(+75.6)
p100k	13	171015	12547	(+92.7)	9	118395	11813	(+90.0)	6	78930	10328	(+86.9)	3	39465	8096	(+79.5)
p141k	20	82400	16372	(+80.1)	15	61800	15549	(+74.8)	9	37080	13271	(+64.2)	5	20600	10377	(+49.6)

of required test frequencies is halved in most cases when targeting $cov = 99\%$. Trivially, the lower the coverage target is, the fewer test frequencies and pattern-configurations are required. For $cov = 90\%$, only very few test frequencies are required to achieve the target coverage with the provided pattern set.

VI. Conclusion

This work presents an approach to predict early life and aging failures by reusing programmable in-situ delay monitors for small- and hidden delay fault detection in Faster-than-At-Speed Test (FAST). It can substantially improve the detection of performance degradation even at lower test frequencies, and provides the possibility to detect target faults with less number of required FAST frequencies. The approach is complemented by a two-step optimization method, which utilizes zero-one linear programming to optimally select frequency and pattern-configurations for generating test schedules with minimal test time. Experimental results show a significant increase in HDF coverage by up to 191% and a reduction of the test time down to 1.5% compared to the conventional case without optimization.

Acknowledgements

This work is part of the project grant WU 245/19-1 funded by the German Research Foundation (DFG). We would like to thank Sebastian Brandhofer for the support regarding the ILP solver.

References

[1] X. Li, J. Qin, and J. B. Bernstein, "Compact Modeling of MOSFET Wearout Mechanisms for Circuit-Reliability Simulation," *IEEE Trans. on Device and Materials Reliability (TDMR)*, vol. 8, no. 1, pp. 98–121, Mar. 2008.

[2] Y. M. Kim, T. W. Chen *et al.*, "Gate-Oxide Early-Life Failure Identification using Delay Shifts," in *Proc. 28th VLSI Test Symp. (VTS)*, Apr. 2010, pp. 69–74.

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

[4] M. Omaña, D. Rossi *et al.*, "Low Cost NBTI Degradation Detection and Masking Approaches," *IEEE Trans. on Computers*, vol. 62, no. 3, pp. 496–509, Mar. 2013.

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

[6] M. Saliva, F. Cacho *et al.*, "Digital Circuits Reliability with In-Situ Monitors in 28nm Fully Depleted SOI," in *Proc. Conf. on Design, Automation Test in Europe (DATE)*, Mar. 2015, pp. 441–446.

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

[8] 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)*, Sep. 2003, pp. 105–111.

[9] M. Amodeo and B. Cory, "Defining faster-than-at-speed delay tests," *Cadence Nanometer Test Quarterly eNewsletter*, vol. 2, no. 2, May 2005.

[10] N. Ahmed, M. Tehranipoor, and V. Jayaram, "A Novel Framework for Faster-than-at-Speed Delay Test Considering IR-drop Effects," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design (ICCAD)*, Nov. 2006, pp. 198–203.

[11] J. Lee and E. J. McCluskey, "Failing Frequency Signature Analysis," in *Proc. IEEE Int'l Test Conf. (ITC)*, Oct. 2008, pp. 1–8, Paper 5.2.

[12] 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, Paper 6.2.

[13] S. Jin, Y. Han *et al.*, "Unified capture scheme for small delay defect detection and aging prediction," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 5, pp. 821–833, May 2013.

[14] C. Liu, M. A. Kochte, and H.-J. Wunderlich, "Aging Monitor Reuse for Small Delay Fault Testing," in *Proc. 35th VLSI Test Symp. (VTS)*, Apr. 2017, pp. 1–6.

[15] 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 Symp. (VTS)*, Apr. 2010, pp. 57–62.

[16] 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)*, Oct. 2014, pp. 1–8, Paper 29.3.

[17] C. Liu, E. Schneider *et al.*, "Extending Aging Monitors for Early Life and Wear-Out Failure Prevention," in *Proc. IEEE 27th Asian Test Symp. (ATS)*, Oct 2018, pp. 92–97.

[18] V. Iyengar, B. K. Rosen, and J. A. Waicukauski, "On Computing the Sizes of Detected Delay Faults," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 9, no. 3, pp. 299–312, 3 1990.

[19] J. A. Waicukauski, E. Lindbloom *et al.*, "Transition fault simulation," *IEEE Design Test of Computers*, vol. 4, no. 2, pp. 32–38, Apr. 1987.

[20] E. Schneider, M. A. Kochte *et al.*, "GPU-Accelerated Simulation of Small Delay Faults," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 36, no. 5, pp. 829–841, May 2017.

[21] K. J. Nowka, G. D. Carpenter *et al.*, "A 32-bit PowerPC System-on-a-Chip With Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling," *IEEE Journ. of Solid-State Circuits (JSSC)*, vol. 37, no. 11, pp. 1441–1447, Nov. 2002.

[22] Cypress Semiconductor Corp., *CY274X, 4-PLL Spread-Spectrum Clock Generator*, Apr. 2017. [Online]. Available: http://www.cypress.com/documentation/datasheets/cy27410-4-pll-spread-spectrum-clock-generator

[23] M. Kampmann, M. A. Kochte *et al.*, "Optimized Selection of Frequencies for Faster-Than-at-Speed Test," in *Proc. IEEE 24th Asian Test Symp. (ATS)*, Nov. 2015, pp. 109–114.

[24] Nangate Inc. (2018) NanGate 45nm Open Cell Library. [Online]. Available: http://www.nangate.com/

[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, Paper 26.1.