Synthesis of Workload Monitors for On-Line Stress Prediction

Rafal Baranowski, Alejandro Cook, Michael E. Imhof, Chang Liu and Hans-Joachim Wunderlich
Institute of Computer Architecture and Computer Engineering, University of Stuttgart, Germany
{baranowski, cook, imhof, liu}@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de

***Abstract*—Stringent reliability requirements call for monitoring mechanisms to account for circuit degradation throughout the complete system lifetime. In this work, we efficiently monitor the stress experienced by the system as a result of its current workload. To achieve this goal, we construct workload monitors that observe the most relevant subset of the circuit's primary and pseudo-primary inputs and produce an accurate stress approximation. The proposed approach enables the timely adoption of suitable countermeasures to reduce or prevent any deviation from the intended circuit behavior. The relation between monitoring accuracy and hardware cost can be adjusted according to design requirements. Experimental results show the efficiency of the proposed approach for the prediction of stress induced by Negative Bias Temperature Instability (NBTI) in critical and near-critical paths of a digital circuit.**

***Keywords*-Reliability estimation, workload monitoring, aging prediction, NBTI**

I. Introduction

Reliability has become one of the most important non-functional properties of safety critical systems, e.g. in medical, aerospace and automotive application domains. This places not only stringent quality requirements on manufacturing test, but also calls for additional monitoring mechanisms throughout the complete system lifetime. These monitoring schemes have to detect any hardware degradation as a result of latent defects, aging or harsh environmental effects, like elevated temperature and electromagnetic interference.

Traditional approaches for reliability monitoring measure *degradation effects*. Therefore, they are useful only after the chip starts to deviate from its intended behavior. For instance, sensors for threshold voltage [1], frequency [2] and slew-rate [3] are successfully applied to estimate the health of semiconductor devices. The critical paths of a digital circuit are monitored with delay sensors, which detect either the violation of a guard-band [4–6], or the failure of a component that is designed to degrade faster than the critical path [7, 8].

In this work, our goal is to quantify reliability risk factors *before* any measurable degradation effect takes place. This enables prediction of reliability issues and adoption of relevant countermeasures. To this end, we estimate the *stress* that a circuit experiences during its operation.

Stress depends on several physical parameters, like chip temperature and supply voltage, as well as on the workload or logic state of the device. While physical properties can be measured directly with appropriate sensors, workload monitoring still remains an open challenge. To address this problem, we present hardware structures which effectively monitor application workload and estimate stress.

As Fig. 1 shows, the *stress monitor* is composed of two main units: The *workload monitor* observes the logic state of a circuit and provides an instantaneous stress estimation. The *stress evaluator* aggregates the output of the workload monitor, together with that of any available on-chip sensors. The evaluator provides the cumulative stress suffered by the circuit over the recent period, e.g. by integration or calculation of a moving average. This fine-grained monitoring approach enables the timely application of any available preventive technique, like load balancing or frequency and voltage scaling, in order to make the system more resilient to stress and less prone to degradation.

This paper details the synthesis of general-purpose workload monitors which can be employed for various stress mechanisms. The accompanying stress evaluator has to be tailored to a specific stress mechanism and is not discussed here in detail. As shown in Fig. 1, a workload monitor is a combinational circuit whose inputs correspond to a subset of the circuit's primary inputs (PI) and pseudo-primary inputs (PPI). The complete set of PI and PPI are collectively referred to as *observables*. Let the number of PI and PPI be n and m, respectively. The workload monitor reads in $p \leq n+m$ observables and provides an instantaneous approximation of a stress metric. Since only PI and PPI nodes are observed and buffered, the proposed monitoring technique has only minimal impact on the mission logic. As the original design is not modified, the proposed method can be applied for IP cores and fixed macros. Note that we do not consider our monitors as a replacement for on-chip sensors; they complement the capabilities of available sensors in order to enable timely prediction and avoidance of reliability problems.

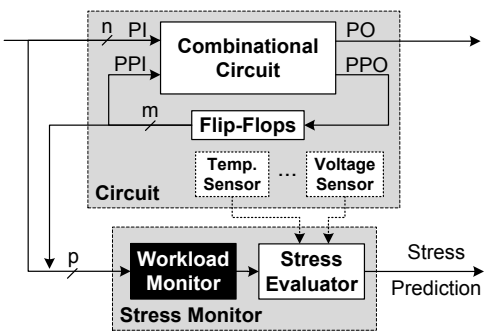

Fig. 1. Monitoring of workload-induced stress

To the best of our knowledge, we present the first general hardware-based approach for the estimation of workload-induced stress. The most similar technique available in the literature is the on-line monitoring of the Architectural Vulnerability Factor (AVF) [9], devised to predict the vulnerability of a microprocessor to soft-errors. In comparison, our approach is more general: The proposed method is applicable to arbitrary digital circuits and can be used to monitor various kinds of failure mechanisms including, for example, AVF and aging induced by Negative Bias Temperature Instability (NBTI).

The rest of this paper is organized as follows: In the next section, we present a methodology for on-line workload monitoring and a stochastic approach for monitor synthesis. Section III presents an application of our method for NBTI aging monitoring and Section IV shows the corresponding experimental evaluation.

II. Construction of Workload Monitors

Our goal is to construct a workload monitor that approximates the stress experienced by a digital hardware component during regular system operation. This monitor is a combinational circuit that reads in the state of observables and provides an instantaneous stress approximation. As on-line observation of all PIs and PPIs is generally infeasible, we search for a small subset of observables which is highly correlated with the target stress mechanism. This subset is then monitored on-line for accurate stress approximation.

We describe the behavior of a workload monitor with an algebraic decision diagram (ADD) [10] referred to as *approximation diagram*. The approximation diagram provides the stress approximation for any assignment of monitor inputs. This diagram is constructed in an iterative procedure: In each iteration, we extend the diagram to improve approximation accuracy. When the accuracy requirements are met, the monitoring circuitry is synthesized from the constructed diagram.

In this section, we describe the monitor construction method. In Section III, we present an application of this method to monitoring of NBTI-induced stress.

A. Stress Metric

The logic state of a digital circuit is defined by the assignment to its PIs and PPIs. The state space is defined as $\{0,1\}^{n+m}$, where n and m are the numbers of circuit's PIs and PPIs, respectively.

We define a stress metric S for a digital circuit as a mapping of its logic state to a real-valued stress measure: $S : \{0,1\}^{n+m} \to \mathbb{R}$. Our definition of the stress metric is very general and can be applied to various instantaneous stress mechanisms which depend on the logic state of the circuit. Examples of such stress mechanisms include logic vulnerability to soft errors [11] or NBTI aging. For instance, in Section III we define an NBTI stress metric as the number of transistors on the critical path that suffer from NBTI-induced stress.

B. Approximation Diagram

A workload monitor implements a stress approximation function $\hat{S} : \{0,1\}^p \to \mathbb{R}$ which maps each assignment of p observables to a real-valued stress approximation, where $p \leq n+m$. Each approximation is an average stress metric over 2^{n+m-p} logic states.

The approximation diagram is a rooted tree (V, E), where $E \subset V \times V$, that represents the function $\hat{S} : \{0,1\}^p \to \mathbb{R}$. Let $T \subseteq V$ be the set of terminal nodes. We define vertex labeling $l : V \to \mathtt{PI} \cup \mathtt{PPI} \cup \mathbb{R}$ as a function that maps each vertex $v \in V$ to either an observable or a real-valued stress approximation:

$$l(v) := \begin{cases} \mathtt{observable} \in \mathtt{PI} \cup \mathtt{PPI} & \text{for } v \in V \setminus T, \\ \mathtt{stress} \in \mathbb{R} & \text{for } v \in T. \end{cases}$$

Each vertex $v \in V \setminus T$ has two outgoing edges in E, a 0-edge and a 1-edge, which specify the successor of v according to the assignment of observable $l(v)$: The 0-edge corresponds to the case when the state of observable $l(v)$ is 0, and the 1-edge corresponds to the case when $l(v)$ is 1. The successor of a vertex connected with a 0-edge (1-edge) is referred to as 0-successor (1-successor). The path π from the root vertex v_0 to a terminal node $v_n \in T$ is the sequence of vertices $v_i \in V$ such that $(v_i, v_{i+1}) \in E$ for $i \in \{0, 1, \ldots, n-1\}$:

$$\pi(v_0, v_n) := (v_0, v_1, \ldots, v_n).$$

Multiple vertices in V may share the same label, but the vertices in every path $\pi(v_0, v_n)$ have unique labels. Each path $\pi(v_0, v_n)$ corresponds to a single assignment of observables defined by the edges connecting consecutive vertices, while $l(v_n)$ provides the real-valued stress approximation for this assignment. The *depth* of the approximation diagram is defined as the maximum number of non-terminal vertices on any path.

As an example, Fig. 2a presents an approximation for the function $\hat{S}(a,b,c)$ defined by the table beside the diagram. 1-edges are represented with a solid line, and 0-edges are dashed. For a given assignment to the observables, the value of function $\hat{S}$ is determined by tracing a path from the root vertex to a terminal node, following the edges that correspond to the given assignment of observables. The function value is obtained from the label of the terminal node. For instance, the stress approximation for $abc = 011$ is 3.0.

C. Diagram Construction Procedure

The problem of diagram construction is formulated as follows: Given a digital circuit, the specification of a stress metric, and the accuracy requirements, construct an approximation diagram that estimates the stress metric for an arbitrary workload (application) with sufficient accuracy.

In order to construct the monitor, we search for observables that are highly correlated with the stress metric and hence are good candidates for on-line observation. To allow for arbitrary workload, we do not make any assumptions about the application and analyze the correlations in Monte Carlo experiments with random input patterns. Alternatively analytical methods can be employed to estimate the signal probabilities [12, 13]. The approximation diagram is constructed with an iterative procedure. In each iteration, the diagram is extended with an additional level of vertices (i.e. additional observables are

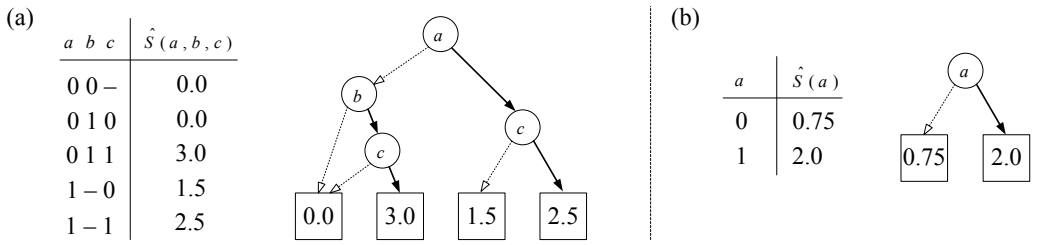

Fig. 2. Example of two stress approximation functions and their corresponding approximation diagrams, with (a) a depth of 3, and (b) a depth of 1

considered) to improve approximation accuracy. The procedure terminates as soon as monitor's accuracy requirements are met.

In the first iteration, we create an approximation diagram with a single (root) vertex connected to two terminal nodes, as shown in Fig. 2b. The root vertex is labeled with an observable that exhibits the *highest correlation* to the stress metric. The terminal nodes connected by the 0- and 1-edge are labeled with the *average* stress metric for the case when the observable is 0 and 1, respectively. Both the correlation analysis and the calculation of the average stress metric is performed in Monte Carlo simulation experiments, as explained in the next section.

In each following iteration, the depth of the approximation diagram is increased by one level: For each path from the root vertex v_0 to a terminal node $v_t \in T$, we find an observable that exhibits the highest correlation to the stress metric when the other observables are fixed to the assignment of $\pi(v_0, v_t)$. Each terminal node is replaced with a new vertex labeled with the observable with highest correlation. At this point, two terminal nodes are added as 0- and 1-successors to each newly created vertex. For each path from the root vertex v_0 to a newly created terminal node v'_t, the terminal node v'_t is labeled with the average stress metric for the case when the observables are fixed to the assignment of $\pi(v_0, v'_t)$. The procedure is repeated until a user-specified bound on the diagram depth is reached, or until the accuracy requirements are met, as explained in Section II-E.

For instance, in order to find the 0-successor of the root node in Fig. 2a, we perform the correlation analysis with observable a fixed to 0. As observable b exhibits the highest correlation to the stress metric when $a = 0$, the 0-successor of a is labeled with b. The terminal node connected to vertex b with the 0-edge is labeled with the average stress metric calculated for the case when ab are constantly assigned 00.

D. Correlation Analysis

Given a candidate observable a and a stress metric S, the correlation coefficient between the observable and the stress metric, denoted by $C(a, S)$, is calculated as a Pearson product-moment correlation coefficient:

$$C(a,S) := \frac{1}{N-1} \sum_{i=1}^{N} \left(\frac{a_i - \overline{a}}{s_a} \right) \left(\frac{S_i - \overline{S}}{s_S} \right)$$

where:

- N is the number of simulated random patterns,
- a_i and S_i are the state of the observable (0 or 1) and the value of the stress metric for the i-th simulated pattern,
- $\overline{a}$ and $\overline{S}$ are the average values of the observable (i.e. signal probability) and the stress metric,
- s_a and s_S are the sample standard deviations of the observable and the stress metric, respectively.

The observable that exhibits the highest correlation to the stress metric is considered for on-line observation, i.e. assigned to a vertex in the approximation diagram.

E. Evaluation of Accuracy

The approximation accuracy is evaluated for the termination criterion during the construction of the approximation diagram, and to analyze the final quality of the predictor.

The approximation error is defined as the difference between the exact stress metric S and the approximation $\hat{S}$. As the evaluation of the error for the entire state space of a circuit is usually unfeasible, Monte Carlo simulation experiments are performed. In each experiment, we simulate a single random pattern, evaluate the exact stress metric S and derive its approximation $\hat{S}$ from the approximation diagram.

We consider three types of approximation errors: maximal error $E_{\texttt{MAX}}$, mean error $E_{\texttt{MEAN}}$, and root mean squared error $E_{\texttt{RMS}}$:

$$\begin{aligned} E_{\texttt{MAX}} &= \texttt{MAX}_{i=1}^{N} |\hat{S}_i - S_i| \\ E_{\texttt{MEAN}} &= \frac{1}{N} \sum_{i=1}^{N} |\hat{S}_i - S_i| \\ E_{\texttt{RMS}} &= \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\hat{S}_i - S_i \right)^2} \end{aligned}$$

where S_i and $\hat{S}_i$ are the exact and approximated stress metrics for the i-th simulated pattern, and N is the number of Monte Carlo experiments.

F. Monitor Synthesis

The final approximation diagram constitutes the behavioral model of the workload monitor. Each path from the root vertex to one of the terminal nodes corresponds to one assignment of observables, while the label of the terminal node provides the stress approximation. The diagram is transformed into a hardware description language and synthesized with a tool for combinational logic synthesis.

The hardware overhead of the monitor depends on the number of levels in the approximation diagram, and the precision of stress approximations. To trade off monitor area for accuracy, real-valued stress approximations are quantized. The smallest change in the stress metric that is measurable by the monitor is referred to as *quantization step size*. As shown in the experimental results, the quantization step size has a significant impact on both the accuracy of a monitor and its area overhead.

III. Application to NBTI Monitoring

In this section, we exemplarily apply the proposed method to NBTI aging monitoring. We construct a monitor that approximates the number of PMOS transistors which suffer from NBTI stress on the critical and near-critical paths. The monitor can be combined with temperature sensors and used to guide NBTI-aware adaptation, e.g. to prevent that an application causing severe degradation is executed at high temperature.

A. NBTI Stress Modeling

The NBTI effect in PMOS transistors consists in oxide degradation caused by formation of traps. The degradation results in a gradual shift in the threshold voltage, which in turn causes an increased propagation delay. Eventually, the NBTI stress may significantly increase the critical path delay and lead to timing violations [14].

The NBTI-induced degradation occurs when a negative voltage is applied between the gate and source of a PMOS device ($\mathtt{U_{GS}} < 0$). For instance, in a CMOS inverter the PMOS transistor suffers from NBTI stress when the input to the inverter is logic 0. In CMOS gates with stacked PMOS transistors, the stress conditions depend on the state of multiple gate inputs [15].

We define the NBTI stress metric as the number of PMOS transistors that suffer from NBTI degradation on the critical path. More formally, we define a function $S_{\mathtt{NBTI}} : \{0,1\}^{n+m} \to \mathbb{N}$ that maps the state of device's PIs and PPIs to a natural number that reflects the number of PMOS transistors on a critical path suffering from NBTI stress. The value of this function for a certain input pattern is easily found in simulation by counting the number of PMOS transistors on the critical path with $\mathtt{U_{GS}} < 0$. Note that our NBTI stress metric does not depend on any NBTI model and is technology independent.

Using the approach described in Section II, we approximate the NBTI stress metric $S_{\mathtt{NBTI}}$ with a function $\hat{S}_{\mathtt{NBTI}} : \{0,1\}^p \to \mathbb{R}$ that maps the assignment of $p \leq n+m$ observables to an average number of PMOS transistors that suffer from NBTI stress under this assignment.

As the NBTI stress metric is a natural number, a quantization step size of 1 is used for approximations, i.e., the real-valued stress in the approximation diagram is rounded to the nearest integer. The resulting monitor gives an integer approximation of the number of PMOS transistors on the critical path that suffer from NBTI degradation.

B. NBTI Stress for k-longest Paths

As the technology scales, digital circuits become more and more balanced, with many paths that may potentially become critical. To deal with a large number of near-critical paths, we construct a monitor for a cumulative NBTI stress metric for k-longest paths.

Our goal is to monitor the *maximum* number of PMOS transistors subject to NBTI stress on the longest paths. Let $S^i_{\mathtt{NBTI}}$ be the NBTI stress metric of an i-th longest path. The NBTI stress metric for k-longest paths is defined as:

$$S^{\mathtt{MAX}}_{\mathtt{NBTI}} = \mathtt{MAX}^k_{i=1}\left(S^i_{\mathtt{NBTI}}\right)$$

The NBTI stress metric for k-longest paths does not inform which path is suffering the most, but gives the number of transistors under stress for the path that suffers the most. For instance, if we have two near-critical paths with 7 and 10 transistors that are currently aging, the metric is 10.

IV. Evaluation

A. Experimental Setup

The proposed method is evaluated on ITC99 and NXP benchmarks. The Nangate 45nm open cell library [16] is used to synthesize the circuits and monitors. The critical and near-critical paths of the circuits are extracted with a commercial static timing analysis (STA) tool.

For each circuit, we build NBTI monitors for the critical path that limits the performance of the circuit, as well as for 10 and 100 paths that may become critical due to aging. During the construction of approximation diagrams, we simulate 20 000 random patterns to find the successors of each vertex (cf. Section II-C).

The monitoring accuracy and its area overhead is analyzed for various depths of the approximation diagram (cf. Section II-B) and different quantization step sizes (Section II-F). The accuracy is evaluated in 20 000 Monte Carlo experiments using the metrics defined in Section II-E.

B. Accuracy of single path monitoring

Table I presents the accuracy and area overhead of NBTI monitors for single critical path. The monitors approximate the number of PMOS transistors on the critical path that suffer from NBTI. The approximations are rounded to the nearest integer (quantization step size is 1). For each benchmark, we evaluate three monitors with 8, 10, and 12 levels in the approximation diagram.

The first three columns in Table I describe the benchmark circuits, including the name, the average number of PMOS transistors suffering from NBTI stress on the critical path (S_{avg}), and the size of each benchmark. The following three columns give the maximum, mean and root mean squared (RMS) approximation error for the 8-level monitor. The presented error metrics are relative to the average stress metric S_{avg} from column 2. The next two columns state the absolute area of the monitor as well as the area overhead (+%) w.r.t. the benchmark area from column 3. These five columns are

TABLE I: ACCURACY AND AREA OVERHEAD OF NBTI MONITORS FOR ONE PATH AND QUANTIZATION STEP SIZE OF 1

Benchmark			Approximation Diagram Depth: 8					Approximation Diagram Depth: 10					Approximation Diagram Depth: 12				
	S_{avg}	Area	Error [%]			Area		Error [%]			Area		Error [%]			Area	
Name *(1)*	[#] *(2)*	[μm^2] *(3)*	max *(4)*	$mean$ *(5)*	rms *(6)*	[μm^2] *(7)*	[+%] *(8)*	max *(9)*	$mean$ *(10)*	rms *(11)*	[μm^2] *(12)*	[+%] *(13)*	max *(14)*	$mean$ *(15)*	rms *(16)*	[μm^2] *(17)*	[+%] *(18)*
b14	48.9	4140	10.2	1.7	2.4	194	+4.7%	10.2	1.6	2.3	643	+15.5%	10.2	1.5	2.2	2129	+51.4%
b15	30.9	6493	16.2	1.7	2.5	92	+1.4%	16.2	1.5	2.3	487	+7.5%	16.2	1.2	2.1	2047	+31.5%
b17	27.6	21058	18.1	2.6	3.6	293	+1.4%	14.5	2.1	3.1	1020	+4.8%	14.5	1.7	2.7	3043	+14.5%
b18	62.9	58830	9.5	1.1	1.9	67	+0.1%	9.5	1.0	1.9	295	+0.5%	9.5	1.0	1.9	1234	+2.1%
b19	51.2	117150	11.7	1.2	2.2	94	+0.1%	11.7	1.2	2.2	348	+0.3%	11.7	1.2	2.1	1335	+1.1%
b20	52.3	8645	13.4	2.2	2.9	138	+1.6%	13.4	2.2	2.8	484	+5.6%	11.5	2.2	2.8	1748	+20.2%
b21	47.5	8762	16.8	2.6	3.4	294	+3.4%	14.7	2.2	3.0	1008	+11.5%	12.6	2.0	2.8	2578	+29.4%
p35k	37.2	17966	13.4	2.7	3.6	206	+1.1%	13.4	2.6	3.5	748	+4.2%	13.4	2.5	3.3	2837	+15.8%
p45k	25.3	18798	15.8	2.1	3.1	217	+1.2%	11.9	1.8	2.8	733	+3.9%	11.9	1.4	2.5	2610	+13.9%
p77k	127.1	29521	5.5	1.1	1.4	328	+1.1%	5.5	1.0	1.3	1220	+4.1%	5.5	0.9	1.2	3560	+12.1%
p81k	59.0	64231	11.9	1.9	2.6	238	+0.4%	10.2	1.8	2.4	851	+1.3%	10.2	1.7	2.3	2741	+4.3%
p89k	38.4	44523	18.2	3.0	3.9	244	+0.5%	18.2	2.8	3.7	925	+2.1%	18.2	2.7	3.5	2725	+6.1%
p100k	25.0	45056	16.0	3.0	4.1	225	+0.5%	16.0	2.7	3.8	921	+2.0%	16.0	2.5	3.5	2976	+6.6%
p141k	27.6	81148	10.9	1.8	2.7	174	+0.2%	10.9	1.6	2.5	640	+0.8%	10.9	1.3	2.3	2014	+2.5%
p239k	63.4	126082	17.4	2.2	3.4	203	+0.2%	15.8	2.1	3.3	669	+0.5%	15.8	2.1	3.2	2346	+1.9%
p259k	65.3	162242	15.3	2.1	3.2	163	+0.1%	15.3	2.0	3.2	581	+0.4%	16.8	2.0	3.1	1838	+1.1%
p267k	27.3	101822	18.3	3.1	4.2	152	+0.1%	18.3	3.0	4.1	448	+0.4%	14.6	2.9	4.0	1314	+1.3%
p269k	26.5	102001	18.9	3.3	4.4	153	+0.2%	15.1	3.2	4.3	559	+0.5%	15.1	3.1	4.2	1789	+1.8%
p279k	31.5	127297	12.7	1.3	2.2	105	+0.1%	12.7	1.1	2.1	394	+0.3%	12.7	1.0	1.9	1281	+1.0%
p286k	35.0	169555	14.3	1.9	2.7	183	+0.1%	14.3	1.6	2.4	710	+0.4%	11.4	1.3	2.1	2482	+1.5%
p295k	46.4	127161	21.6	3.1	3.9	201	+0.2%	19.4	2.9	3.8	727	+0.6%	19.4	2.8	3.6	2410	+1.9%
p330k	22.2	133719	9.0	1.3	2.5	105	+0.1%	9.0	0.9	2.1	352	+0.3%	13.5	0.6	1.7	1074	+0.8%

TABLE II: ACCURACY AND AREA OVERHEAD OF NBTI MONITORS FOR ONE PATH AND QUANTIZATION STEP SIZE OF 2

Benchmark			Approximation Diagram Depth: 8					Approximation Diagram Depth: 10					Approximation Diagram Depth: 12				
	S_{avg}	Area	Error [%]			Area		Error [%]			Area		Error [%]			Area	
Name	[#]	[μm^2]	max	$mean$	rms	[μm^2]	[+%]	max	$mean$	rms	[μm^2]	[+%]	max	$mean$	rms	[μm^2]	[+%]
b15	30.9	6493	19.4	2.1	3.0	42	+0.7%	19.4	2.0	2.8	271	+4.2%	19.4	1.9	2.6	1113	+17.1%
b18	62.9	58830	9.5	1.1	2.0	23	+0.0%	9.5	1.1	2.0	143	+0.2%	9.5	1.1	1.9	778	+1.3%
b20	52.3	8645	13.4	2.3	3.0	65	+0.7%	13.4	2.3	3.0	219	+2.5%	13.4	2.3	2.9	992	+11.5%
p35k	37.2	17966	16.1	2.9	3.8	118	+0.7%	16.1	2.8	3.7	393	+2.2%	13.4	2.7	3.6	1694	+9.4%
p77k	127.1	29521	6.3	1.1	1.5	202	+0.7%	6.3	1.1	1.4	593	+2.0%	5.5	1.0	1.3	2470	+8.4%
p89k	38.4	44523	20.8	3.1	4.1	127	+0.3%	20.8	3.0	3.9	450	+1.0%	18.2	2.9	3.8	1547	+3.5%
p141k	27.6	81148	10.9	2.3	3.2	107	+0.1%	10.9	2.2	3.1	413	+0.5%	10.9	2.1	2.9	1240	+1.5%
p259k	65.3	162242	16.8	2.4	3.4	117	+0.1%	15.3	2.3	3.3	280	+0.2%	16.8	2.3	3.2	1156	+0.7%
p269k	26.5	102001	18.9	3.6	4.8	77	+0.1%	18.9	3.5	4.6	306	+0.3%	18.9	3.4	4.5	1001	+1.0%
p286k	35.0	169555	14.3	2.2	3.0	74	+0.0%	14.3	2.0	2.8	392	+0.2%	11.4	1.8	2.6	1124	+0.7%
p330k	22.2	133719	13.5	2.5	3.6	126	+0.1%	13.5	2.4	3.4	316	+0.2%	13.5	2.4	3.3	625	+0.5%

repeated for the remaining two monitors, with 10 and 12 levels in the approximation diagram.

For approximation diagrams with the depth of 8, the maximum error (column 4) ranges between 5.5% and 21.6%. This means that in the worst case (p295k), there exists an input pattern, for which the approximation provided by the monitor differs from the exact value by 21.6%. The mean error is significantly lower: In the worst case (p269k), the approximation differs from the exact value by 3.3% on average. The RMS error is bound to a maximum of 4.4% (p269k): Assuming that the approximation error is normally distributed, the error is within $\pm$8.8% for 95% of patterns.

If more levels are considered in the approximation diagram, the approximation accuracy improves: For the largest circuit (p330k), the RMS error is 2.5% with 8 levels, 2.1% with 10 levels, and 1.7% with 12 levels. A similar improvement is observed for other large benchmarks.

Table II shows the results for a quantization step size of 2. The table includes every second benchmark from Table I. Compared to the results with a quantization step size of 1, the approximation error increases but is still within $\pm$10% for 95% of random patterns (RMS error is below 5% for all benchmarks).

C. Accuracy of k-longest paths monitoring

Table III presents the accuracy for 10- and 100-longest path monitoring, for every third benchmark from Table I. For some benchmarks, such as p330k, the more paths are monitored, the more levels in the approximation diagram are required to preserve the accuracy. However, the monitors for other benchmarks become more accurate when more paths are monitored (e.g. b18). In the latter circuits, the maximum stress metric for multiple paths is subject to less variation than the

TABLE III: ACCURACY AND AREA OVERHEAD OF NBTI MONITORS FOR 10 AND 100 LONGEST PATHS AND QUANTIZATION STEP SIZE OF 1

	Benchmark			Approximation Diagram Depth: 8					Approximation Diagram Depth: 10					Approximation Diagram Depth: 12				
		S_{avg}	Area	Error [%]			Area		Error [%]			Area		Error [%]			Area	
	Name	[#]	[μm^2]	max	$mean$	rms	[μm^2]	[+%]	max	$mean$	rms	[μm^2]	[+%]	max	$mean$	rms	[μm^2]	[+%]
10 Paths	b14	54.7	4140	11.0	1.8	2.5	202	+4.9%	9.1	1.6	2.4	741	+17.9%	9.1	1.6	2.3	2284	+55.2%
	b18	71.8	58830	7.0	0.4	1.0	64	+0.1%	7.0	0.4	0.9	211	+0.4%	7.0	0.4	0.9	717	+1.2%
	b21	51.4	8762	9.7	1.9	2.4	128	+1.5%	9.7	1.8	2.4	523	+6.0%	9.7	1.8	2.4	1752	+20.0%
	p77k	131.5	29521	6.8	1.0	1.4	251	+0.8%	6.8	1.0	1.3	989	+3.3%	6.8	1.0	1.3	3021	+10.2%
	p100k	32.9	45056	15.2	2.5	3.3	244	+0.5%	12.2	2.3	3.1	852	+1.9%	12.2	2.1	2.9	2936	+6.5%
	p259k	68.6	162242	10.2	1.4	2.1	126	+0.1%	10.2	1.3	2.1	405	+0.2%	10.2	1.3	2.0	1654	+1.0%
	p279k	55.1	127297	9.1	1.7	2.3	240	+0.2%	9.1	1.6	2.2	742	+0.6%	9.1	1.5	2.1	2767	+2.2%
	p330k	22.7	133719	13.2	1.4	2.6	117	+0.1%	13.2	1.0	2.1	308	+0.2%	8.8	0.7	1.7	950	+0.7%
100 Paths	b14	55.5	4140	10.8	1.6	2.3	204	+4.9%	9.0	1.5	2.2	720	+17.4%	9.0	1.4	2.1	2107	+50.9%
	b18	74.3	58830	5.4	0.0	0.3	7	+0.0%	5.4	0.0	0.3	21	+0.0%	5.4	0.0	0.3	77	+0.1%
	b21	56.6	8762	10.6	1.7	2.3	223	+2.5%	8.8	1.6	2.2	720	+8.2%	8.8	1.5	2.2	2191	+25.0%
	p77k	135.9	29521	6.6	1.1	1.4	249	+0.8%	5.9	1.0	1.4	997	+3.4%	5.9	1.0	1.3	3123	+10.6%
	p100k	36.9	45056	13.6	2.3	3.1	246	+0.5%	13.6	2.2	3.0	912	+2.0%	13.6	2.0	2.8	2850	+6.3%
	p259k	69.0	162242	10.1	1.3	2.0	116	+0.1%	10.1	1.3	2.0	486	+0.3%	10.1	1.3	2.0	1791	+1.1%
	p279k	57.3	127297	10.5	1.7	2.2	237	+0.2%	10.5	1.6	2.1	825	+0.6%	10.5	1.5	2.0	2966	+2.3%
	p330k	35.0	133719	20.0	3.6	4.7	180	+0.1%	20.0	3.5	4.6	581	+0.4%	17.2	3.5	4.5	2286	+1.7%

stress of a single path (i.e. its value depends less on the input patterns). Thus, the maximum stress metric is sometimes easier to approximate than the stress of a single critical path.

D. Hardware Overhead

The area overhead of the workload monitors is shown in Tables I, II and III. It is exponential in the number of levels used in the approximation diagram, and depends on the quantization step size. For the quantization step size of 1, the maximum area overhead is 328, 1220 and 3560 μm^2 for 8-, 10-, and 12-level monitors. For the quantization step size of 2, the area is reduced by up to 66%. As the monitor area depends little on the size of the monitored circuit, the relative cost of monitoring decreases with an increasing circuit size.

V. CONCLUSION

On-line stress estimation has become mandatory for applications with reliability requirements. We propose a novel method to monitor workload-induced stress. We estimate the degradation rate of a circuit by observing its logic state. Our method is suitable for the monitoring of various degradation mechanisms for which the application workload plays an important role. As an example, we apply the technique to monitor the number of transistors on the critical path that suffer from NBTI degradation. Typically, the developed monitors require an area overhead of under 1% and offer an average estimation error below 3.3%.

VI. ACKNOWLEDGMENT

Parts of this work were supported by the German Research Foundation (DFG) under grants WU 245/13-1 (RM-BIST) and WU 245/11-1 (OASIS).

REFERENCES

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

[2] 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.

[3] A. Ghosh, R. Brown, R. Rao, and C.-T. Chuang, "A precise negative bias temperature instability sensor using slew-rate monitor circuitry," in *Proc. IEEE Intl. Symp. on Circuits and Systems (ISCAS)*, 2009, pp. 381–384.

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

[5] J. Vazquez, V. Champac, I. Teixeira, M. Santos, and J. Teixeira, "Programmable Aging Sensor for Automotive Safety-Critical Applications," in *Proc. Design, Automation and Test in Europe (DATE)*, 2010, pp. 618–621.

[6] M. Agarwal, V. Balakrishnan, A. Bhuyan, K. Kim, B. Paul, W. Wang, B. Yang, Y. Cao, and S. Mitra, "Optimized Circuit Failure Prediction for Aging: Practicality and Promise," in *Proc. IEEE Intl. Test Conference (ITC)*, 2008, paper 26.1.

[7] S. Mishra, M. Pecht, and D. L. Goodman, "In-situ sensors for product reliability monitoring," in *Storage and Retrieval for Image and Video Databases*, vol. 4755, 2002, pp. 10–19.

[8] M. Nakai, S. Akui, K. Seno, T. Meguro, T. Seki, T. Kondo, A. Hashiguchi, H. Kawahara, K. Kumano, and M. Shimura, "Dynamic Voltage and Frequency Management for a Low-Power Embedded Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 28–35, 2005.

[9] K. R. Walcott, G. Humphreys, and S. Gurumurthi, "Dynamic Prediction of Architectural Vulnerability From Microarchitectural State," *SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 516–527, Jun. 2007.

[10] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, "Algebraic Decision Diagrams and Their Applications," in *IEEE/ACM Proc. Intl. Conference on Computer-Aided Design (ICCAD)*, 1993, pp. 188–191.

[11] S. Mukherjee, C. Weaver, J. Emer, S. Reinhardt, and T. Austin, "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor," in *Proc. IEEE/ACM Intl. Symposium on Microarchitecture (MICRO-36)*, 2009, pp. 29–40.

[12] K. Parker and E. McCluskey, "Sequential circuit output probabilities from regular expressions," *IEEE Transactions on Computers (TC)*, vol. 27, no. 3, pp. 222–231, 1978.

[13] H.-J. Wunderlich, "Protest: A tool for probabilistic testability analysis," in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, 1985, pp. 204–211.

[14] W. Wang, V. Reddy, A. Krishnan, R. Vattikonda, S. Krishnan, and C. Y, "Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology," *IEEE Transactions on Device and Materials Reliability*, vol. 7, no. 4, pp. 509–517, 2007.

[15] K. K. Saluja, S. Vijayakumar, W. Sootkaneung, and X. Yang, "NBTI Degradation: A Problem or a Scare?" in *Proc. IEEE Intl. Conference on VLSI Design (VLSID)*, 2008, pp. 137–142.

[16] Nangate 45nm Open Cell Library v1.3, http://www.nangate.com.