Multi-Path Ageing Sensor for Cost-efficient Delay Fault Prediction

Gaole Sai, Basel Halak, *Member, IEEE,* and Mark Zwolinski, *Senior Member, IEEE,*

***Abstract*—Aggressive technology scaling has accelerated the susceptibility of CMOS devices to aging effects. Consequently, the speed of a path can degrade significantly over time; this results in delay faults. Dynamic reliability management schemes have been proposed to ensure an IC's lifetime reliability. Such schemes are typically based on the use of aging sensors to predict a circuit's failure before errors actually appear. Existing aging sensors are usually placed on the circuit's longest delay paths, which are deemed to be the most vulnerable to delay faults. However, complex designs typically have a large number of long delay paths that need to be monitored. Such approaches are very costly and may be infeasible. This work proposes a new aging sensor, capable of monitoring multiple paths concurrently. The proposed sensor has been designed at transistor level using a 32nm technology and applied to a 32-bit MIPS to monitor 10 paths concurrently. Our results show that using the proposed sensor for monitoring 10 paths can save 197.1% and 97.1% in area overheads compared to Razor and Canary, respectively.**

***Index Terms*—Ageing Sensor, NBTI, Timing Error.**

I. Introduction

Process variations and aging-induced device degradation are becoming major reliability concerns in modern semiconductor technologies. Both phenomena lead to performance degradation, and hence timing errors. To avoid delay fault induced failures, integrated circuits are typically designed with large safety margins [1]. This generally means a circuit is designed for worst-case operating conditions. Such an approach may limit system performance and lead to an increase in power consumption [2]. Dynamic reliability management schemes have been proposed to assure an IC's lifetime reliability. Such schemes are typically based on the use of sensors to predict circuit failures before errors actually appear. The system can then adaptively scale its operating frequency and supply voltage according to the actual operating conditions to compensate for performance degradation [1].

Various in situ delay monitoring sensors have been proposed. These include delay fault detection and prediction techniques [3], [4]. Existing delay fault sensors are usually placed on the circuit's longest delay paths. However, the increasing complexity of ICs has led to a significant rise in the number of long paths and potential aging-critical paths that may be vulnerable to timing errors [5]. This means the cost of in situ delay monitoring may be prohibitive. The objective of this work is to minimize the area overhead with any increase in potential aging-critical paths.

In this paper, we propose a new Differential Multiple Error Detection Sensor (DMEDS) for timing errors. DMEDS is able to monitor multiple paths simultaneously, which significantly reduces the number of sensors needed to monitor aging-induced delay faults. DMEDS has been designed at transistor level in a 32nm CMOS technology and verified at system level. Our results indicate that the use of the proposed sensor for delay fault monitoring across 10 paths can lead to a significant saving in area overhead compared to Razor [3], and Canary [4]: 197.1%, 97.1%, respectively.

This paper is organized as follows. Section II briefly describes related work. Section III outlines the design principles of DMEDS. Verification results and cost analysis are discussed in section IV. Finally, conclusions are drawn in section V.

The authors are with the Department of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ United Kingdom (email: {g.sai, bh9, mz}@ecs.soton.ac.uk).

II. Background and Previous Work

A. Bias Temperature Instability

Negative Bias Temperature Instability (NBTI) is considered as one of the most important aging reliability issues in PMOS transistors. NBTI occurs because interface traps at the gate oxide interface are generated when a negative gate bias – a negative potential difference between gate and source (Vgs), is applied. It manifests itself as an increase in the threshold voltage (Vth) and a decrease in the drain current, thereby degrading timing. NBTI can cause more than 20% timing degradation in the worst-case operating conditions, which may change the ranking of critical paths, [2]. In order to estimate the behavior of an IC due to NBTI effects, a number models have been proposed for different levels of abstraction [2], [5]. Once ΔV_{th} is estimated, it can be integrated into circuit simulation tools, thus potential NBTI-critical paths can be identified. Positive BTI (PBTI) is the equivalent aging reliability issue in NMOS transistors. The threshold voltage shift of PBTI has become significant in High-K Metal Gate technologies [2], [6]. The double effect of NBTI and PBTI exacerbates the timing degradation of ICs.

B. Existing Sensors

Unlike delay fault detection sensors, such as Razor FF [3], the Canary FF [4], shown in Fig. 1, is a delay fault prediction sensor that checks data consistency before the rising clock edge. As shown in Fig. 1, the shadow FF receives delayed data as a reference and compares it with the data from the main FF. As the path ages, the Error signal will be triggered if the delayed data violates the setup time of the shadow FF. As a delay fault prediction sensor, the Canary FF is more suitable for aging detection compared to Razor. In this case, the input signal may violate the setup time of the shadow FF, but not that of the main FF, as a result of circuit aging. Therefore it does not require any circuitry for timing

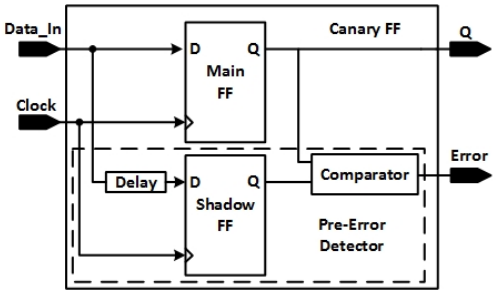

Fig. 1. Canary Flip-Flop [4]

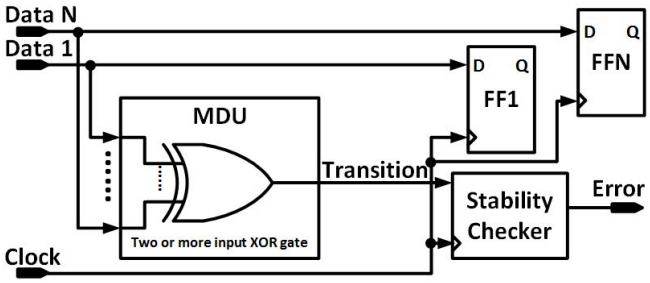

Fig. 2. Architecture of DMEDS

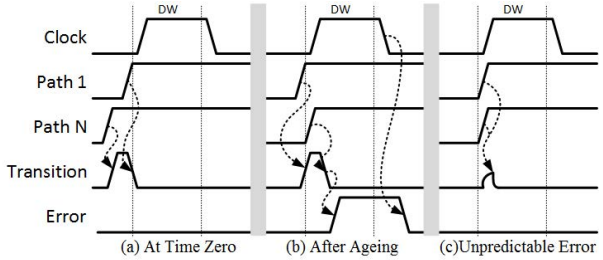

Fig. 3. Timing Diagram for DMEDS

error recovery nor for metastability detection in the main FF. However, metastability occurring in the shadow FF will cause errors. Moreover, implementing a shadow FF of the same size will cause a relatively large area overhead.

A number of latch-type delay fault monitoring sensors have been proposed in recent years, [7], [8]. These replace the main FF with a latch to solve the metastability issue and to decrease the transition time from D to Q. However, replacing the FF in a pipelined system may require extra circuitry to compensate for further reliability issues, as the previous stage and current stage will be connected to each other directly while the clock is at logic '1' [9]. Existing sensors can only detect the timing error for one path, therefore the area cost for multiple path monitoring might be prohibitive, as will be shown later.

III. Differential Multiple Error Detection Sensor (DMEDS)

This section outlines the operating principles of the DMEDS. The advantages of DMEDS compared with existing approaches are:

1) DMEDS is able to predict the delay fault from multiple paths, which improves the cost-efficiency.
2) DMEDS is an external sensor; it does not replace the FF of the original design, and therefore it is easy to implement and will not influence the functionality or performance of the original design.
3) DMEDS is an error prediction sensor. Consequently, DMEDS is able to prevent metastability and predict delay faults when the propagation delay of a path is about to violate the setup time of a FF.

A. Operating Principles of DMEDS

The architecture of DMEDS is shown in Fig. 2. Compared with existing delay fault monitoring sensors, DMEDS retains the original main FF and is able to monitor delay faults from two or more paths at the same time. This can significantly improve the cost-efficiency. As shown in Fig. 2, compared with the Canary FF, DMEDS replaces the Pre-Error Detector circuitry with a Multiple Detection Unit (MDU) and a Stability Checker. Its main advantages over the Canary FF are that it has less area overhead and it does not suffer from the metastability problem. The MDU monitors two or more of the Potential Critical Paths (PCPs) simultaneously. Any transitions in the data will trigger a transition of the MDU output signal (from '0' to '1' or '1' to '0'). This transition will be captured if it occurs during the detection period of the stability checker, thus signaling a delay fault. The MDU includes a prediction margin from its own delay. The stability checker checks the stability of the delayed signal while the clock is '1'. Therefore the detection window of DMEDS starts from T_{MDU} before the rising clock edge until T_{MDU} before the falling clock edge, where T_{MDU} is the delay of the MDU. The prediction margin ensures the signal error triggers before the real delay fault occurs, thus enabling delay fault prediction. The width of the prediction margin at time zero can be adjusted by scaling the transistors in the MDU or by adding buffers [10]. DMEDS might generate an incorrect error signal if other short paths share the same end as the PCPs. Buffers and clock duty cycle adjustment are required to ensure all the transitions at the monitoring point respect the detection window, [8], before and after aging.

In practice, it is very unlikely that the data from two or more different paths converge at exactly same time. The MDU assumes that there will be timing differences between the transitions from different paths. The XOR gate(s) in the MDU compares the data monitored by the DMEDS and each transition of the input data will trigger a transition at the MDU output. However, the sensitivity of the MDU circuit is not sufficiently high to detect an extremely small difference between transitions of the input data. Transitions become undetectable when an even number of signals changes at about the same time. In reality, transitions from PCPs will not be 100% correlated, as shown in section III-C. Transitions will be detected by the MDU eventually when an odd number of PCPs are assessed. As the DMEDS is a delay fault prediction sensor, it is not necessary to detect every single transition in the MDU as the circuit ages, provided that some are detected before a delay fault occurs in a PCP.

The stability checker checks the stability of the output signal from the MDU. It captures transitions of the MDU output signal during the checking period, from the rising edge until the falling edge of the clock signal. As shown in Fig. 3, output signal 'Error' will be triggered if the signal 'Transition' changes during the checking period and is cleared after the falling clock edge. There are three typical operating cases during the DMEDS operation time, when the DMEDS is monitoring two or more PCPs simultaneously, as shown in Fig. 3 (a), (b) and (c). Paths '1' and 'N' are chosen to illustrate the operation, but the principle applies to any pair of paths and to situations where 'Path 1' ages more than 'Path N'.

(a) The propagation delay of 'Path N', is smaller than that of 'Path 1'. As the paths have been optimized by static timing analysis tools, the remaining slacks in any of those paths should not allow transitions to reach the Detection Window (DW) at time zero, and the error signal is not triggered.

(b) After a certain period of time, both 'Path 1' and 'Path N' age, but as 'Path N', is aging much faster, the remaining slack in 'Path N' shrinks to the detection window duration before that in 'Path 1' (and other paths). The output signal of the MDU is activated by the transition in 'Path N' within the detection period of the stability checker. This transition is then captured and triggers the error signal.

(c) In this case, 'Path N', is aging only slightly faster than 'Path1', so the difference between the changes from those paths becomes smaller and smaller during the operation time. After a certain period of time, the remaining slack of both paths approaches the DW and the difference between 'Path 1' and 'Path N' is not recognized by the MDU. A small glitch is generated when 'Path 1' and 'Path N' change at about the same time. The glitch is too small to be captured by the stability checker. The delay faults of both paths become unpredictable. This situation is most likely to arise because of correlation between paths and can be avoided by selecting different paths for monitoring (see section III-C).

From previous work, a timing monitoring sensor would not be implemented on every path. A limited set of paths should be identified [11]. A PCP may become *the* critical path after fabrication due to process variations, or after a certain time because of aging. Therefore, identifying the critical path and the PCPs at time zero is required as a part of the timing analysis. The detection window and setup time of the FFs will be degraded after aging. If the setup time degrades faster than the detection window because of device aging, the detection window must also be adjusted, according to worst case aging and process variations analyses in order to ensure that it is sufficiently wide.

B. Transistor Level Design of DMEDS

The circuit schematics of the MDU and the stability checker are shown in Fig. 4. Fig. 4 (a) shows a 2-input MDU, which is a 6 transistor XOR gate. The stability checker, Fig. 4 (b), is simplified with respect to the stability checkers proposed previously [11]. The stability checker detects whether the input signal is stable. X and Y are the input signals of a NOR gate. During the clock high state – the checking period – node X is pulled down when the input signal is logic '1' and node Y is pulled down when the input signal is logic '0'. The error signal will be triggered when the input signal is both logic '0' and '1' during the same checking period. The checking period begins T_{MDU} before the clock edge. Therefore the prediction margin will be $T_{MDU} - T_{setup} - T_{pd}$, where T_{setup} is the setup time of the FFs at the monitoring point and T_{pd} is the pull down time of node X or Y. Hence a delay fault can be predicted (see Fig. 3). M4 and M6 will pull down nodes X and Y when the signal 'Error' is not a strong logic '1'. This therefore increases the sensitivity of the stability checker. Both nodes X and Y will be pulled up during the clock low state, which clears the error signal. The error signal needs to be stored as the stability checker cannot latch the error signal after the falling edge of the clock signal. The MDU will not be able to generate a strong pulse signal to pull down nodes X or Y if the difference between input signals is smaller than the minimum measurement resolution, as shown in Fig.3 (c). The resolution and prediction margin are adjusted by tuning transistor sizes, [10].

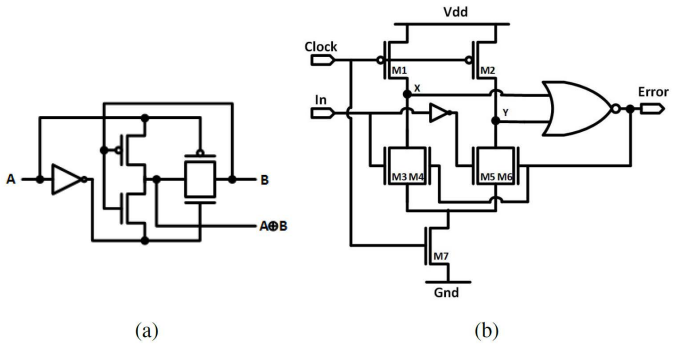

Fig. 4. Transistor Level Design of DMEDS

C. Path Selection

To present the path selection in a DMEDS implementation, we have considered a 32-bit pipelined MIPS. The PCPs were identified by performing a detailed timing analysis using a 32nm technology. The PCPs can be classified as follows:

I. Data is written back to the specific bits of different addresses. The data will not be written to different addresses in the same clock cycle, thus is 100% de-correlated.

II. Data is written back to the specific bits of the same address or read from the register file. The critical path will be active while the data changes from '1' to '0' or '0' to '1'. The PCPs switching rate will be 0.25 if the signal probability is 50%. Fig. 5 shows the transition probability of PCPs, where α is the PCPs switching rate and n is the number of PCPs, according to Equations (1), odd transitions, (2), even, and (3), none, respectively. As Fig. 5 shows, the probability of odd transitions is generally higher than even transitions. The correlation between PCPs increases with the number of PCPs. In practice, the correlation rate over a certain amount of time determines the effectiveness of the delay fault prediction. In the worst case, the probabilities of even and odd transitions are 50% with a 99% confidence level and $\pm 4\%$ interval in

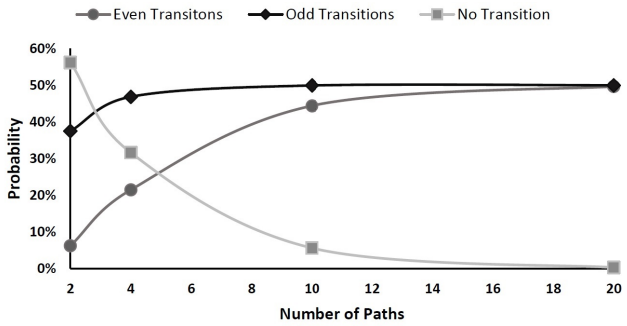

Fig. 5. Percentage of Transitions

every 1000 clock cycles. Therefore, the number of PCPs that are detected by a single DMEDS is dependent on the accuracy constraints and error sampling size of the design.

$$\sum_{k=0}^{i} C(2k+1,n)\alpha^{2k+1}(1-\alpha)^{n-(2k+1)}, (i \leq \frac{n-1}{2}) \quad (1)$$

$$\sum_{k=1}^{i} C(2k,n)\alpha^{2k}(1-\alpha)^{n-2k}, (i \leq \frac{n}{2}) \quad (2)$$

$$(1-\alpha)^{n} \quad (3)$$

III. Data is assigned to the carry chain in the ALU. This might cause a few inputs of the FFs at the end of EX stage to change in the same clock cycle. The correlation rate between transitions will be high in this case, as those paths share the carry chain. However, path sharing means the same work load, and hence the transistor stress of those paths will be about the same. The ranking of those paths is most likely not affected by aging, which means that only one of them needs to be monitored. In some particular cases, the correlated paths should be monitored by different DMEDSs to ensure decorrelation of the signals.

To monitor N paths, the MDU has N-1 2-input XOR gates and the data propagates through $\lceil \log_2 N \rceil$ XOR gates. The increase in N will lead to an increase in the prediction margin, therefore the prediction margin can be larger than the original safety margin, reserved for PVTA variations, if N is large enough. Hence, the number of paths monitored by one DMEDS should be limited according to the application. In Adaptive Voltage Scaling(AVS) approaches, the prediction margin should be slightly larger than the delay degradation of a minimum voltage drop, which ensures the true error will not occur after the voltage scaling and guarantees the system is running at the lowest voltage level. In this case, limiting the number of paths monitored by same DMEDS or inserting a clock buffer for the stability checkers is required. Therefore more than one DMEDS is required for delay fault prediction.

IV. Verification and Comparative Analysis

This section first presents the results of functional verification at transistor level and system level, then we summarize the cost of the proposed delay fault monitoring technique.

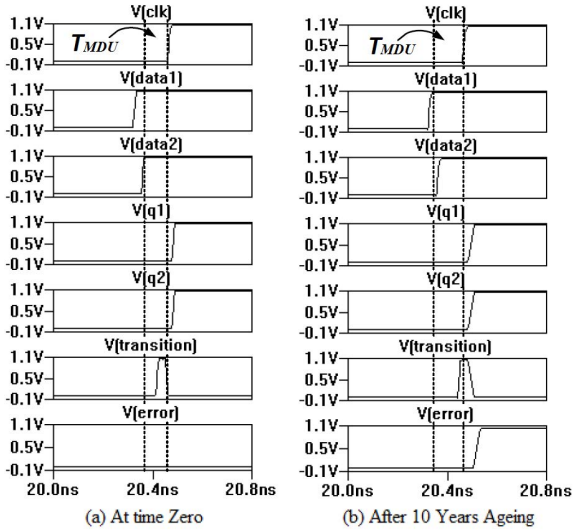

Fig. 6. Multiple Timing Error Detection

A. Transistor Level Simulation Results

DMEDS was designed using a 32nm CMOS technology for verification and evaluation. Fig. 6 shows the Spice simulation results when the inputs of two conventional FFs are monitored simultaneously at time zero and after 10 years. The aging degradations were estimated using Synopsys HSPICE MOSRA. As Fig. 6 (a) shows, the difference between 'data1' and 'data2' triggers a pulse of the output signal 'Transition' at time zero. As the remaining slacks in any of those paths do not reach the DW yet, the error signal is not triggered. After 10 years ageing, the minimum measurement resolution between two input has degraded from 21ps to 39ps. The width of the T_{MDU} and the FF's set-up time are increased by 24.5% and 126%, respectively, due to the aging of the MDU and FF. The simulation results for DMEDS after 10 years' aging is shown in Fig. 6 (b). Both 'data1' and 'data2' switch at the same times as in Fig. 6 (a). A pulse is generated by the MDU during the stability checking period caused by the difference between 'data1' and 'data2'. The stability checker captures the pulse and triggers the error signal due to the aging degradations of the DW. As 'data1' and 'data2' are propagated to the output signals of FFs, 'q1' and 'q2', after the rising clock edge, there is no timing error. According to the worst/best case process variation simulation results, the minimum measurement resolution, T_{MDU} and FF's set-up time vary by +0.4%/-5.0%, +3.2%/-2.2% and +2.3%/-5.8% at time zero, respectively.

B. DMEDS Functional Verification

An equivalent model of DMEDS, written in a hardware description language, has been used to verify the functionality of the DMEDS at system level, as shown in Fig. 7. The behavior of the model is identical to that shown in Fig. 3.

We have used a 32-bit pipelined MIPS to verify the functionality of DMEDS. Fig. 8 shows the MWB stage writing data back to ten different register files. The unbalanced addressing will cause these ten paths to age differently. Transitions in

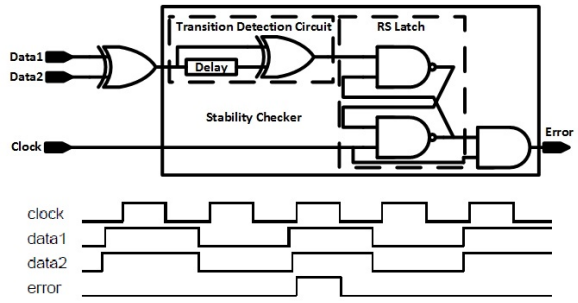

Fig. 7. Equivalent Model of DMEDS and Simulation Result

TABLE I
COMPARISON WITH OTHER DESIGNS

FF Type	Razor [3]	Canary [4]	DMEDS (II)
Number of extra Transistors (I)	54 225%	30 125%	6.7 27.9%
Metastability	Yes	Yes	No
Error Recovery Required	Yes	No	No
Replace FFs	Yes	No	No
Latch Type	**DSTB [7]**	**iRazor [8]**	**DMEDS (II)**
Number of extra Transistors (III)	34 212.5%	9.46 59.1%	6.7 41.9%
Metastability	No	No	No
Error Recovery Required	Yes	Yes	No
Replace FFs	Yes	Yes	No

(I) Compared to standard 24T FF (excluding the delay chain)
(II) 10-input DMEDS monitors 10 path simultaneously (nine 2-input XOR (54T) and one stability checker (13T) shared)
(III) Compared to standard 16T Latch

those ten paths occur successively because they are triggered by the same instructions but by different addresses (100% de-correlated). The error signal is triggered when late transitions are detected by the 10-input DMEDS.

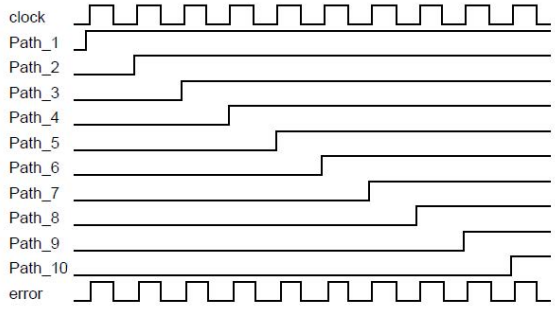

Fig. 8. 10-input DMEDS functionality verification in a 32-bit MIPS

C. Cost Comparison

Table I gives a cost and performance comparison with different delay fault detection or prediction sensors. The overhead of FF type sensors is compared with the standard 24 transistor FF. Compared with Razor and Canary, DMEDS saves 197.1% and 97.1% of the transistors respectively for 10 path delay fault monitoring (10 paths are monitored by one 10-input DMEDS simultaneously). DMEDS is also able to predict delay faults in a latch-type sensor. The overhead of a latch-type sensor is compared with the standard 16 transistor latch, Table I. Compared with DSTB and iRazor, DMEDS saves 170.6% and 17.2% of the transistors respectively for 10 path delay fault monitoring, which significantly reduces the area overhead. The cost of MDU adjustment is not included in this comparison, as existing sensors will also require adjustments to ensure their functionality after serious aging such as the delay chain of Canary FF [4] and the local CLK Generation of iRazor [8]. The routing area overhead has been carefully considered in this comparison. Compared with other designs, DMEDS has more routing area overhead on the input side. However, there is only one error signal on the output side, while there is a higher routing area overhead on the output side for the other designs. Furthermore, more circuitry will be required to manage the error signals when those signal delay fault sensors are implemented. Thus, the overhead of the other sensors are underestimated compared with DMEDS.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a new sensor, named DMEDS, for predicting timing errors. Compared with Razor [3], Canary [4], DSTB [8], DMEDS saves at least 197.1%, 97.1%, 170.6% and 17.2% of the transistors, respectively, which significantly reduces the area overhead. The design in a 32 nm technology was verified at transistor level and at system level. Future research will focus on cell library design and implementing an AVS system with DMEDS on a high-performance processor to assess system-level trade-offs.

REFERENCES

[1] S. Agwa *et al.*, "ERSUT: A self-healing architecture for mitigating PVT variations without pipeline flushing," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 11, pp. 1069–1073, 2016.
[2] E. Mintarno *et al.*, "Self-tuning for maximized lifetime energy-efficiency in the presence of circuit aging," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst*, vol. 30, no. 5, pp. 760–773, 2011.
[3] S. Das *et al.*, "A self-tuning DVS processor using delay-error detection and correction," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, 2006.
[4] H. Fuketa *et al.*, "Adaptive performance compensation with in-situ timing error predictive sensors for subthreshold circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*, vol. 20, no. 2, pp. 333–343, 2012.
[5] A. Calimera *et al.*, "Design techniques for NBTI-tolerant power-gating architectures," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 4, pp. 249–253, 2012.
[6] Z. C. Lee *et al.*, "NBTI/PBTI-aware WWL voltage control for half-selected cell stability improvement," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 9, pp. 602–606, 2013.
[7] K. A. Bowman *et al.*, "A 45 nm resilient microprocessor core for dynamic variation tolerance," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 194–208, 2011.
[8] Y. Zhang *et al.*, "iRazor: 3-transistor current-based error detection and correction in an ARM Cortex-R4 processor," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*. IEEE, 2016, pp. 160–162.
[9] Y. Huang *et al.*, "Computation-skip error mitigation scheme for power supply voltage scaling in recursive applications," *Springer J. Signal Process Sys*, pp. 1–12, 2016.
[10] M. Alioto *et al.*, "General strategies to design nanometer flip-flops in the energy-delay space," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1583–1596, 2010.
[11] J. Semiao *et al.*, "Performance sensor for tolerance and predictive detection of delay-faults," in *IEEE Int. Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, 2014, pp. 110–115.