Timing Management Strategies: Comparing the use of Recovery, Masking, and Borrowing Strategies

Navid Khoshavi Najafabadi

navid.khoshavi@knights.ucf.edu

Department of Engineering and Computer Science,
University of Central Florida

Abstract

Although today’s trends of technology scaling is going to bring higher performance computer systems, it also results in integrated circuits (ICs) suffering from increasing Process, Voltage, and Temperature (PVT) variations and adverse aging effects. In most cases, these reliability threats manifest themselves as timing errors on critical speed-paths of the circuit, if a large design guard band is not reserved. In this work, three categories of resilient design methodologies for timing errors are introduced and their ability to recovery error-state is evaluated. In addition, we propose an architectural approach consists of detection and recovery phases to recover circuit from BTI-induced aging. The BTI impact on the critical and near critical paths performance is continuously examined through a small logic circuit which is responsible to assert error signal in the case of any timing violation in those paths. By declaring timing violation occurrence in the system, the timing sensitive portion of the circuit is recovered from BTI through switching computations to replicate critical and near critical paths and turning off the primary critical and near critical paths. The proposed technique achieves aging mitigation and reduced energy consumption as compared to a baseline circuit which mitigates aging by adding significant voltage guardbands to meet the desired timing specification.

Key words: Dynamic Voltage Scaling, Dynamic Power Consumption, Timing Error Masking, Redundant Logic Block, Approximate circuit, Time Borrowing, Adaptive Time Borrowing

I. Introduction

With technology scaling, integrated circuits suffer from increasing Process, Voltage, and Temperature (PVT) variations and aging effects [1],[2], [3], [4]. Conventional circuit designs tolerate these variations by embedding a large timing guard-band into the design to ensure error-free computing [2], [3], [5], [6], [7]. Unfortunately, such conservative design methodology reduces the benefits provided by technology scaling due to imposing high performance and energy overhead and increasing the testing time. Consequently, resilient design methodologies for timing errors have emerged as alternative solutions which can be shown as Fig. 1 and can be categorized as the following:

Timing Error Recovery: The prior works achieve timing error resilient through double-sampling latches to detect timing errors on circuit critical path. Razor [7] is the most well-known technique in this area. Once an error is detected, Backward Error Recovery (BER) is conducted to restore the system to an error-free state.

Timing Error Masking: RAZOR-like techniques suffer from the error-free state checkpointing overhead. So, the in-situ timing error correction techniques that are able to mask errors without any rollback are favorable. The timing error masking techniques are fitted in the Forward Error Recovery (FER) category in which a redundant logic block is added to overwrite the outputs of the circuit upon application of inputs that sensitize critical paths [8], [9], [10], [11]. With such exact sensitization constraint, the redundant error-masking circuit tends to have more timing slack when compared to the original circuit, and hence is immune to timing errors.

Time Borrowing: For those flip-flops (FFs) driven by circuit critical paths, denoted as suspicious FFs (SFFs), if they are followed by noncritical paths, they can be replaced with sequential elements having time-borrowing capability and correct timing errors by delaying the arrival time of the correct data to the next logic level [9], [12].

Fig. 1. Taxonomy or resilient design methodologies for timing errors
II. Timing Error Recovery using Razor

The key idea of Razor [7] is to purposely operate the circuit at sub-critical voltage and tune the operating voltage by monitoring the error rate. This eliminates the need for conservative voltage margins. The trade-off would be between the power penalty incurred from error correction against the additional power savings obtained from operating at a lower supply voltage. Operating at moderately sub-critical voltages causes only a few critical instructions to fail. If the non-zero error rate is maintained sufficiently low, then the power overhead from correcting these errors is minimal, while the power savings from the reduced operating voltage can be substantial. This approach is called timing speculation methodology.

A. Detecting Circuit Timing Errors with Razor Flip-Flops

As shown in Fig. 2, in order to detect an error at the circuit level, each flip-flop is augmented by a shadow flip-flop, which is clocked by a delayed clock. If the combinational logic meets the setup time of the main flip-flop, then the main and delayed flip-flops will latch the same value. In this case, the error signal remains low. If the setup time of the main flip-flop is not met, then the main flip-flop will latch a value that is different from the shadow flip-flop. To guarantee that the shadow flip-flop always latches the input data correctly, the input voltage is constrained such that under the worst-case condition, the logic delay does not exceed the shadow flip-flop’s setup time.

The operation of the error detection technique is illustrated in the timing diagram in Figure 2 (b). In the first clock cycle, input data instr 1 meets the setup time of the main flip-flop and shadow flip-flop, thus both flip-flops latch instr 1. But in the second clock cycle, the input data instr 2 does not satisfy the setup time requirement of the main flip-flop. It latches the old data instr 1 while the shadow latch latches instr 2. This is detected by the XOR circuit and the error signal is asserted.

The asserted error signal is used in error correction, where it triggers the correct output value from the shadow latch to be restored to the main flip-flop in the subsequent cycle. This value is then available to the next pipeline stage. Many noncritical flip-flops in a design will not need Razor technology. For example, if the maximum delay at a flip-flop input is guaranteed to meet the required cycle time under the worst-case sub-critical voltage setting, it isn’t necessary to replace it with a Razor flip-flop because it will never need to initiate timing recovery.

B. Recovering Pipeline State afterTiming-errorDetection

A pipeline recovery mechanism guarantees that any timing failures that occur will not corrupt the register and memory state with an incorrect value. Two approaches have been developed to recover pipeline state. The first is a simple but low throughput method based on clock gating, while the second method is a much more scalable technique based on counterflow pipelining.

Fig. 3 illustrates pipeline recovery using a global clock-gating approach. If any pipeline stage detects an error, pipeline control logic stalls the entire pipeline for one cycle by gating the next global clock edge. The additional clock period allows every stage to recompute its result using the Razor shadow latch as input. Since all stages reevaluate their result, any number of errors can be tolerated in a single cycle and forward progress is guaranteed.

Clock gating is not useful in aggressively clocked designs because of its effect on processor cycle time. Instead, a fully pipelined error recovery mechanism based on counterflow pipelining techniques can be used. Fig. 4 illustrates this approach, which places negligible timing constraints on the baseline pipeline design at the expense of extending pipeline recovery over a few cycles. When a Razor flip-flop generates an error signal, pipeline recovery logic must take two specific actions. First, it generates a bubble signal to nullify the computation in the following stage. This signal indicates to the next and subsequent stages that the pipeline slot is empty. Second, recovery logic triggers the flush train by asserting the ID of the stage generating the error signal. In the following cycle, the Razor flip-flop injects the correct value from the shadow latch data back into the pipeline, allowing the errant instruction to continue with its correct inputs.

Additionally, the flush train begins propagating the failing stage’s ID in the opposite direction of instructions. At each stage that the active flush train visits, a bubble replaces the pipeline stage. When the flush ID reaches the start of the pipeline, the flush control logic restarts the pipeline at the instruction following the failing instruction. In the event that multiple stages
generate error signals in the same cycle, all the stages will initiate recovery, but only the failing instruction closest to the end of the pipeline will complete. Later recovery sequences will flush earlier ones.

C. Supply Voltage Control

The voltage control system adjusts the supply voltage based on the monitored error rates. It works to maintain a constant small non-zero error rate. This optimal value depends on factors like cost of error recovery and performance requirements. At regular intervals the error rate of the system is measured. If the error rate is very low, it could indicate circuit computation is finishing too quickly and voltage should be lowered. On the other hand, an increase in the error rate indicates failing timing constraints, and voltage should be increased. The optimal error rate depends on a number of factors including the energy cost of error recovery and overall performance requirements, but in general it incurs a small non-zero error rate.

Fig. 5 illustrates the Razor voltage control system. The control system works to maintain a constant error rate of $E_{diff}$ at regular intervals the error rate of the system is measured by resetting an error counter which is sampled after a fixed period of time. The computed error rate of the sample $E_{sample}$ is then subtracted from the reference error rate to produce the error rate differential $E_{diff}$. $E_{diff}$ is the input to the voltage control function, which sets the target voltage of the voltage regulator. If $E_{diff}$ is negative the system is experience too many errors, and voltage should be increased. If $E_{diff}$ is positive the error rate is too low and voltage should be lowered. The magnitude of $E_{diff}$ indicates the degree to which the system is “out of tune”.

Experimental results have shown that the power savings on reducing the supply voltage below critical voltage can be significant. For instance, the multiplier circuit equipped by RAZOEO, fails quite gracefully, taking nearly 200 mV to go from the point of the first error (1.54 V) to an error rate of 5% (1.34 V). The increase in error rate with decreasing supply voltage corresponds to an increase in the power required for error correction. Below a particular voltage, this power overhead would be greater than the processing power. Therefore, the optimum voltage would be the one which leads to a balance between the power savings with reduced supply voltage and the power overhead of correction. Figure 6 shows the relationship between the power consumption and the supply voltage.

D. Energy Analysis

Fig. 6 illustrates qualitatively the relationship between supply voltage, total energy and pipeline throughput. The total energy consumed by the processor ($E_{total}$) is the sum of the energy required to perform processing operations ($E_{proc}$) plus the energy required to recover the pipeline in the event of a processor timing error ($E_{recovery}$). Moreover, there is a fixed amount of energy overhead incurred to implement Razor. This energy is consumed by the shadow latches and comparison logic. A trade-off exists between the processor and recovery energy components. When supply voltage is decreased, the energy required to perform processing operations is decreased, but fewer of these operations are able to complete within the clock period. As a result, pipeline recovery is invoked more frequently with additional energy expense. Energy for the processor ($E_{proc}$) is optimized when any additional decrease in voltage results in an energy savings that is smaller than the extra energy cost incurred by more pipeline recoveries. The energy-optimal voltage varies from program to program (and even within the phases of a program) because pipeline error rate is heavily dependent on the data values sent to the adder. Under a performance constraint, the optimal voltage is limited to the minimal energy that meets the performance constraint.

Fig 7. shows the energy-optimal supply voltage, average adder error rate, energy reduction, and IPC reduction at the fixed energy-optimal voltage for bzip benchmark (optimal Vdd = 1.1, Error Rate % = 0.31%, Energy Reduced= 57.6%, % IPC Reduced= 0.70%). Clearly, there is 57.6% energy to be reclaimed by running the adder at 0.31% error rate. One particularly encouraging result is that error rates and performance impacts are muted up to and slightly past the energy-optimal voltage, after which error rates rise very quickly.
III. Timing Error Masking Strategy using InTimeFix

The basic idea of the proposed InTimeFix technique [11] is to generate approximate logic for the original logic function of suspicious FFs in such manner that it covers all the logic minterms that sensitize speed-paths. The approximate logic is defined as: Given two Boolean functions $F$ and $G$, $G_0$ is a 0-approximate logic of $F$ if $G_0 = 0 \Rightarrow F = 0$. Similarly, $G_1$ is a 1-approximate logic of $F$ if $G_1 = 1 \Rightarrow F = 1$. According to counter-positive law, two statements can be obtained that are $F = 1 \Rightarrow G_0 = 1$ and $F = 0 \Rightarrow G_1 = 0$. Consider a Boolean function $F = a + b + \overline{a}cd$, there are 13 on-set minterms and 3 off-set minterms in its truth table. Similarly, a 0-approximate logic function of $F$, $G_0 = a + b + \overline{a}c$, covers 2 out of 3 off-set minterms of $F$. Regarding to the Fig. 8 which is constructed circuit $F' = F \cdot G_0 + G_1$, $P$ is all the minterms in $F$’s truth table, $P_0$ is the off-set minterms of $F$ covered by $G_0$ and $P_1$ is the on-set minterms of $F$ covered by $G_1$.

Generally speaking, since the approximate logic circuit is much simpler when compared to the original circuit, its computational latency is smaller. A suspicious FF is driven by multiple paths, and timing errors may occur only when speed-paths are sensitized. In other words, timing errors may be activated by only a few minterms of the truth table for a suspicious FF, denoted as critical minterms. Motivated by this observation, if all the critical minterms are covered with approximate logic, a large timing slack is achieved and potential timing errors is masked.

A. Timing Error Masking with Approximate Logic

An example circuit shown in Fig. 9 for explanation, wherein path $P$ (Input1, A, D, H, F, G, I, J) is a speed-path. When logic ‘1’ is applied at Input1 and propagates along this path to generate logic ‘0’ at the receiving end, we have to assign logic ‘1’ at the side-input of gate A. This is because, logic ‘1’ is a non-controlling value of AND gate, and the output of A will be dominated by the side-input if it is assigned with controlling value. Similarly, side-inputs of gate F and gate I have to be assigned as non-controlling values (see Fig. 9). Side-outputs on the path is defined as the input that need to have deterministic non-controlling values (marked in shade) as essential side-inputs. Based on the path sensitization theory: To cover all the critical minterms that sensitize a particular speed-path is equivalent to approximate its essential side-inputs.

With the above, redundant approximate logic can be constructed for each speed-path by simple structural analysis. Again, take path $P$ in Fig. 9 as an example. In order to construct 0-approximate logic for this path, firstly, the entire path is duplicated and then gradually those gates without essential side-inputs are removed. While the outputs of gates $F$ and $A$ are determined by

---

Fig. 5. Supply Voltage Control System [6]

Fig. 6. Razor power saving [7]

Fig. 7. Relative process energy and pipeline throughput for BZIP [7]
both on-input and side-input signals, two gates need to be duplicated in the 0-approximate logic, and the side-inputs are connected to the same net as path $P$ in the original circuit. As shown in Fig. 9, the 0-approximate logic constructed as above will output logic ‘0’ if and only if the speed-path $P$ in the original circuit is sensitized with launching value logic ‘1’. The 1-approximate logic for speed-path $P$ can be constructed similarly (see Fig. 9).

B. Experimental Results

When comparing the Worst Case Delay (WCD) between the original circuit and the one equipped with redundant approximate logic (see Table 1, Columns 7-10), it can be observed that the proposed solution is able to achieve 11.15% timing slack relaxation on average. On the other hand, as shown in Columns 5-6, the hardware cost (the unit of cost is the area of smallest 2 input AND gates) introduced in the proposed InTimeFix technique to achieve the above timing slack is extremely low, less than 0.89% on average. As can be seen from Column 11, the runtime to process the largest benchmark circuit ethernet takes less than one second. Consequently, the proposed methodology can be easily scalable to large industrial designs.

IV. Time Borrowing using Dadgour’s method

This work [12] offers two major contributions to the aging-resilient circuit design methodology literature. First, it introduces a novel sensor circuit that can detect the aging of pipeline architectures by monitoring the arrival time of data signals at flip-flops. In order to screen the aging process, conventional methods try to measure the delay of digital circuits. Although this circuit can accurately measure the degradation of circuits, it needs a considerable amount of hardware overhead. In this work, an effective and at the same time, low-overhead and low-power circuit is proposed for the early detection of aging in nanoscale circuits. The area overhead of the proposed circuit is estimated to be less than 45% compared to that of previous approaches, which are over 95%. To ensure the accuracy of its operation, a comprehensive timing analysis is performed on the proposed circuit including the influence of process variations. As a second contribution, this work presents an innovative correction technique to reduce the probability of timing failures caused by aging. This method employs novel reconfigurable flip-flops, which operate as normal flip-flops as long as the circuit is fresh, but function as time-borrowing flip-flops once the circuit ages. This unique flip-flop design allows utilization of the advantages of the time-borrowing technique while avoiding potential race conditions that can be created by employing such a technique.

A. Sensor Circuit

The proposed circuit is composed of two stages as shown in Fig. 10. Stage #1 generates a “voltage glitch” on its output node, F, if the circuit is aged; otherwise, F remains low. Since a glitch is not a valid logic value, Stage #2 is designed to convert the glitches on F to stable logic levels on OUT signal.

A schematic diagram presenting signal waveforms at different nodes of Fig. 10 is shown in Fig. 11 to demonstrate its basic operation. In this figure, it is assumed that the inputs of flip-flops (D) are connected to the outputs of CLBs of the pipeline architecture. The output of Glitch Generator (F) becomes high only if all three inputs of the NAND gate (X, Y and Q) are ‘1’.

Table 1. Experimental results on improved timing slack and hardware cost [11]

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Circuit Size (# of gates)</th>
<th>FF (#)</th>
<th>Critical FF (#)</th>
<th>Cost (# of gates)</th>
<th>Improved Ratio (%)</th>
<th>InTimeFix WCD (ns)</th>
<th>Related Slack (ns)</th>
<th>Improved Ratio (%)</th>
<th>Runtime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c38417</td>
<td>24370</td>
<td>1636</td>
<td>78</td>
<td>570</td>
<td>2.24</td>
<td>35.34</td>
<td>31.95</td>
<td>3.41</td>
<td>9.64</td>
</tr>
<tr>
<td>v38584</td>
<td>21066</td>
<td>1436</td>
<td>12</td>
<td>98</td>
<td>0.47</td>
<td>20.50</td>
<td>18.49</td>
<td>2.01</td>
<td>9.80</td>
</tr>
<tr>
<td>des_perf</td>
<td>154233</td>
<td>9105</td>
<td>89</td>
<td>592</td>
<td>0.38</td>
<td>7.80</td>
<td>6.68</td>
<td>1.12</td>
<td>14.36</td>
</tr>
<tr>
<td>wb_connax</td>
<td>75352</td>
<td>3316</td>
<td>277</td>
<td>1160</td>
<td>1.54</td>
<td>8.47</td>
<td>7.74</td>
<td>0.73</td>
<td>8.59</td>
</tr>
<tr>
<td>ethernet</td>
<td>157841</td>
<td>10752</td>
<td>28</td>
<td>346</td>
<td>0.24</td>
<td>8.21</td>
<td>7.11</td>
<td>1.10</td>
<td>13.38</td>
</tr>
<tr>
<td>Ave.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.89</td>
<td></td>
<td></td>
<td></td>
<td>11.15</td>
</tr>
</tbody>
</table>
When the CLB is fresh, its output (D) arrives well before the clock edge and as a result, output F remains low since at any given instant, at least one of the inputs of the NAND is low (Fig. 11 (a)). On the other hand, if signal D arrives late enough due to the aging of the CLB, there will be a time interval where all three inputs of the NAND gate are ‘1’ which generates a “voltage glitch” at the output node F (Fig. 11 (b)).

The proposed sensor is more area- and power-efficient compared to the one presented in [13] the pre-sampling-based circuit. The area and power overhead associated with these sensors are compared and results are shown in Fig. 12. In this figure, “overhead” refers to any circuit component other than the original flip-flop that is added to the circuit to measure the late arrival of signals at the input of that flip-flop. As it can be observed, the proposed circuit has clear advantages in terms of having minimal design overheads. Area and power savings offered by the proposed sensor become even more significant when a large number of sensors are deployed in VLSI designs. Furthermore, the impact of various levels of process variations on these sensors is investigated in Fig. 12. Here, it is assumed that the parameter fluctuations affect all transistors in the designs. One can observe that the detection error rates (number of faulty detections to total detections) are comparable for all three sensors.

B. Correction Technique Using Reconfigurable Flip-Flops

Once the performance degradation due to aging is detected, proper actions should be taken in order to prevent timing failures. In this work, a novel adaptive “time-borrowing” approach is proposed to reduce the probability of timing violations in aged circuits. This is achieved by replacing the original flip-flops in the pipeline architectures with reconfigurable flip-flops. The conventional pipeline designs cause only a small number of Hold time violations when the circuit is fresh; however, such circuits have higher probability of Setup time violations after the aging occurs. On the other hand, the time-borrowing (TB) method reduces the probability of the Setup time failures after Combinational Logic Block (CLB) ages. Nevertheless, such an approach can cause repeated Hold time violations when the circuit is fresh. In this work, an adaptive time-borrowing approach (Adaptive TB) is proposed that can combine the advantages of both the conventional and time-borrowing pipelining techniques. In this scheme, the circuit employs a conventional pipelining method before the aging of CLBs, which results in low Setup and Hold failure probabilities. After CLBs age (which can be detected by the proposed sensor), the pipeline system switches to a time-borrowing architecture, thereby resulting in low Setup and Hold failure probabilities. The proposed adaptive time-borrowing approach employs “reconfigurable flip-flops”, which operate as conventional flip-flops when the CLBs are fresh and perform as time-borrowing flip-flops when the CLBs are aged. The reconfigurable flip-flops are designed by introducing minor modifications to the design of traditional time-borrowing flip-flops. In order to understand how the reconfigurable flip-flops function, one needs to understand the basic operation of time-borrowing flip-flops. The gate-level representations of the conventional, time-borrowing and reconfigurable time-borrowing flip-flops are shown in Fig. 13. These flip-flops employ identical master/slave latch architectures (Fig. 13). However, they differ in the clocking schemes that have been used to drive the clock signals of their master and slave latches (Fig. 13 (b)-(d)).

C. Experimental Results

It can be observed that the conventional techniques results in a large number of Setup time violations (Fig. 14) when the CLBs are aged while the time-borrowing approach suffers from high Hold time failures (Fig. 14 ) when the CLBs are fresh. The proposed adaptive time-borrowing technique, however, effectively eliminates such timing failures in all cases. In order to investigate the impact of process variations, the effectiveness of three pipelining approaches are evaluated in terms of their failure probabilities (which are defined as the highest failure probability considering both the Setup and Hold time failures over the life-time of the circuit). The failure probabilities are shown in Fig. 15 for various process variation levels.
If we assume inverters have independent Gaussian delay distribution \((\mu, \sigma)\), the delay of a path including \(n\) inverters obey the Gaussian distribution \((n\mu, \sqrt{n}\sigma)\). The Dadgour’s method tolerates better the failure probability with the increase of process variation in compare to two other techniques. It can be observed that the likelihoods of timing failures for all pipelining techniques increase approximately at the same rate by escalating the parameter fluctuations. Furthermore, the timing failure probabilities are investigated for deeper pipeline architecture with various numbers of pipeline stages as shown in Fig. 15. It is shown that although the timing violations increase dramatically for the conventional and the time-borrowing techniques, the failure probability for the proposed approach increases at a smaller rate than two other methods. Hence, the proposed technique can be more effective for deeper pipelines.

V. Proposed Approach to Rejuvenate BTI-Inducing Aging of Critical Path in CMOS Logic

If the increased delay due to BTI, not appropriately provisioned, timing failures on critical logic paths may happen. To recover circuit from BTI degradation in the case of timing violation on critical path, we proposed remodeling critical path to meet time constraints and putting the timing critical portions of the circuit on sleep mode for the stress relaxation purpose. It has been shown in [19] that there is a significant spread between the length of the critical path and the majority of paths. Specifically, over 95% of the logic paths exhibit less than 75% of the length of the longest logic path. On the other hand, there may exist some other near critical paths which become critical during circuit lifetime due to varying level of stress. Thus, replicating only a single critical path may not be sufficient. For our experiments, we only select the top 10% of critical paths for replication and protection against cumulative delay variations due to aging effects.

A. Aging-aware Dispatcher for Representative Critical Path Selection

The RAZOR method [7] is used to detect timing violation through sampling the output data of critical path at two different points in time. The earlier, speculative sample is stored in a data latch which is called main latch. This main latch is augmented by a “shadow latch” which is clocked by a delay clock. Consequently, if any timing violation happens due to BTI, the value of one of main and shadow latches will be changed and detected by an Exclusive-OR gate which compares output of main and shadow latches at the negative edge of the clock. Fig. 16 (a) shows timing diagram in which the main and shadow latches capture the same value. Therefore, the error signal remains low. Fig. 16 (b) illustrates timing diagram when BTI increase delay of the circuit. In cycle 1, the combinational logic exceeds the intended delay due to BTI. So, both main and shadow latches capture input data \(D_{in}\), but in the second clock cycle, transition change of input data \(D_{in}\) will be captured by shadow latch while main latch still keeps old \(D_{in}\). By comparing the valid data of the main and shadow latches, an error signal is then generated in cycle 2.

B. Remodeling of Critical Path for Aging Mitigation

The error signal generated by Exclusive-OR gate is used to put the critical path on the sleep mode and active redundant critical path. Thus, circuit can recover from BTI while it is on sleep mode. The proposed idea has been shown in Fig. 17. It shows a timing sensitive portion of circuit consists of both critical and near critical paths has been replicated. In order to detect
timing violation in the circuit, only the critical path and near critical paths of each instance are equipped by a RAZOR.

When the RAZOR detects any timing violation in the either critical path or near critical paths, it produces an error signal which we call it here switch signal. The switch signal is connected to clock input of a positive edge-triggered D flip-flop. This means that the flip flops can only change output values when the switch signal is at a positive edge. The input signal D is fed by Redundant Sleep Transistor (RST) control signal which comes from the inverted Primary Sleep Transistor (RST) control signal. In the beginning, the RST signal equals to ‘1’ which means the redundant critical path is on Sleep mode. Subsequently, PST signal is ‘0’ which means Critical Path and near Critical Paths functions as usual. When timing violation happens in the primary critical path, this circuit is turned off and disconnected from the VDD through the sleep. Simultaneously, redundant critical path is turned on.

C. Switching Interval Impact on the BTI Recovery

The proposed approach is a on-demand BTI recovery method which switches autonomously between two critical paths throughout the operational period of the circuit to keep overall delay degradation to a minimum. Only one instance is active at any given time which can be implemented by generation of mutually-exclusive control signals for the sleep transistors.

The switching interval determines the maximum permitted degradation incurred by a set of critical paths. Fig. 18 shows the effect of automatic switching for C880 and frg2 benchmarks such that the delay of the circuits is constrained to a timing specification of 1% in the first day of circuit lifetime. A proper delay reduction is achieved by using the redundant critical path and autonomous switching interval. The critical path of C880 benchmark degrades faster than the critical path of frg2, subsequently, it needs more switching times over a specific operation time.

VI. Experimental Results

The proposed methodology is evaluated by simulation of C880, i5 and frg2 circuits from MCNC benchmark suite. Our
circuit-level modeling is performed via Synopsys HSPICE [16] reliability analysis to simulate BTI effect for the 45nm Nangate open cell library. In order to apply aging, we used MOSRA model. The MOSRA model is constructed with physics-based formulations and augmented with coefficient parameters, to improve the model accuracy and parameter extraction flexibility. HSPICE reliability analysis includes two simulation phases which are the fresh simulation phase and the post-stress simulation phase. In fresh simulation phase, HSPICE computes the electron stress of selected transistors in the circuit based on the circuit behavior and HSPICE built-in stress model of BTI effect. According to the stress information produced during the fresh simulation phase, HSPICE simulates the degradation effect on the circuit performance in post-stress simulation phase.

NBTI/PBTI has two phases depending on bias condition of PMOS/NMOS transistor: stress phase and recovery phase. In the stress phase, threshold voltage \( V_{th} \) of PMOS/NMOS increases. During the recovery phase, \( V_{th} \) degradation is partially recovered. In order to calculate \( V_{th} \) shift, we used built-in model provided by MOSRA [16] in which two principal physical mechanisms are considered: One related to the contribution of the interface traps (Equation 1) and the other related to the traps deep inside the dielectric layer (Equation 2).

\[
\Delta V_{th,\text{st}} = \exp\left[-\frac{E_b}{E_{tr}}\left(V_{th} - V_{th,0}\right)\right] \cdot \exp\left[TITFD \cdot E\left(V_{gs}, V_{ds}\right) \right]^{not}
\]

\[
\Delta V_{th,\text{ot}} = \exp\left[-\frac{TOTTD}{E\left(V_{gs}, V_{ds}\right)}\right]^{not}
\]

In the above equations, \( E(V_{GS}, V_{DS}) \) denotes the strength of the electric field of the dielectrics. Regarding to the significant dependence of NBTI and PBTI on the channel length, flexible channel width- and length-dependence equations are included in the BTI model of MOSRA. The partial-recovery effect is modeled by taking into account the stress stimulus duty cycle. When the partial-recovery effect is considered, the total degradation becomes smaller:

\[
\Delta V_{th,\text{ac}} = TTD0 \cdot \Delta V_{th} \cdot \exp\left(-TDCD, g\right)
\]

where the \( g \) quantity models the effect of duty cycle.

For power-gating, the sleep transistors are instantiated in the header and considered to provide superior leakage power reduction. The size of the sleep transistors are set such that the voltage drop across them is around 1% of supply voltage [18]. To quantify the benefit of the proposed method, two set of experiments is devised such that the delay of the circuits is constrained to a timing specification of 1% and 2%. The supply voltages for the proposed method are selected based on the amount of permitted degradation at the design time. For example, considering the specified voltage for baseline circuit be 1.1V, the rise time of critical path of C880 benchmark is 409 picoseconds in the initial time of circuit operation. Suppose 1% degradation is permitted for the circuit. This means we can adjust the initial voltage of the circuit such that the rise time of critical path of C880 benchmark starts from 405 picoseconds in the beginning of circuit lifetime. Consequently, if the critical path of this circuit degrades more than 1%, redundant critical path is activated and put primary critical path on sleep mode. Table 2 shows the supply voltage setting and switching interval for C880, i5 and frg2 benchmarks for 1% and 2% permitted degradation. The number of switching between critical path and redundant critical path has been decreased over a long period of circuit operation by allowing the circuit to be degraded more. This is because the supply voltage of circuit is established greater than permitted 1% degradation to tolerate 2% degradation in the initial time of circuit operation.

VII. Future Works

In the future works, we will report the energy benefit in terms of area cost. The potential benefit of the proposed technique is to be quantified in terms of total lifetime energy reduction as compared to a baseline circuit which mitigates aging by adding voltage guardbands to meet the desired timing specification. This means that the proposed technique achieves aging mitigation and reduced energy consumption. As we expected, none of the baseline circuits operating at nominal voltage would have met the timing specification. Thus, the supply voltage is needed to be increased. Operation at this elevated voltage also result in an increased rate of degradation whereas, the proposed method delay degradation for all benchmarks are below the permitted amount of degradation with reference to the delay of baseline circuit operating at nominal voltage in the initial time of circuit operation.

VIII. Conclusions
A comprehensive comparison between presented techniques is demonstrated in Table 3. The error detection of Razor is based on duplication of latch/FF while InTimeFix uses redundant logic in order to detect error. Dadgour’s method utilizes low-overhead sensor in order to detect errors. Razor and Dadgour’s methods impose large sequential overhead to the circuit while InTimeFix method imposes more combinational overhead. Finally, all three techniques achieve full timing margin recovery. The proposed technique uses RAZOR method to detect any timing violation in the either critical path or near critical path. Hence, the flip-flop duplication, clock-tree loading and high sequential overhead are inherited features from RAZOR technique. On the other hand, the critical and near critical paths are replicated in the proposed method which results in imposing moderate combinational overhead. However, the rollback or instruction replay is not necessary in the proposed approach in order to recover errors. Consequently, the proposed technique can be considered as a forward error recovery technique.

References
DATE, pp. 244 - 249, 2010.


