Time Borrowing Flip-Flop architecture for error masking in near threshold voltage regime

Abhinav Raj, Diwakar Arora, Aman Chaurasia, Prof. S. Indu
Delhi Technological University, New Delhi, India

rajabhina586@gmail.com

Abstract. This As dynamic delay variability increases with near-threshold voltage operation, wide timing margins need to be integrated in DVFS (Dynamic Voltage and Frequency Scaling) when deciding the frequency of operation to ensure reliable and error-free operation. Hence the maximum frequency (which is already low) in near-threshold operation is limited by these timing margins. Online methodologies for the resilience of temporal arrangement errors facilitate the recovery of temporal arrangement margins, increasing efficiency and/or energy consumption. This study presents a method for the resilience of online temporal arrangement errors that masks temporal arrangement errors by borrowing time from the phases of serial pipelines. Planned style, while not instruction replay or roll-back support, will recover temporal arrangement margins.

1. INTRODUCTION

VLSI has advanced substantially over the past few years. Energy consumption by transistors is hardly affected by reductions in supply voltages or by doubling the transistor count in every successive generation. There is a dire need to substantially reduce the energy utilization in order to achieve energy efficiency. The minimum requisite for the aforementioned is the sizeable pruning of energy utilization per operation. There are 3 regions of operations in any MOSFET device based on supply voltage, as are presented in Table 1.

Near-threshold voltage (NTV) can be understood simply by evaluating the crest of minimum energy and subsequently measuring the service voltage at that location. In near-threshold region, a substantial decrease in energy can be noticed without a significant performance degradation. Hence, this region is favorable for applications which requires high efficiency at modest speed.

As dynamic delay variability increases with near-threshold voltage operation, it is necessary to use large timing margins in DVFS (Dynamic Voltage and Frequency Scaling) when deciding the frequency of operation to ensure reliable and error-free operation. Hence the maximum frequency (which is already low) in near-threshold operation is limited by these timing margins.

One of the key merits of a timing error resilient machine is its ability to profoundly enhance the efficacy of the system, refine its power consumption and most importantly, aid in recovery of timing margins. In this paper, we seek to propose a timing error resilience strategy that mitigates any timing errors by time borrowing from sequential pipelining stages. Unaccompanied by any instruction replay or roll-back support, the proposed model will recover timing margins.

The paper progresses as follows: Section 2 describes prior work done in field of timing error resilience. Section 3 talks about the design challenges in the near threshold region and Timing
violations are errors in Flip Flops. Section 4 talks about timing violations and errors in Flip Flops. Section 5 puts forth the incentive for the error resilience circuit. Section 6 talks about Latches for error masking and critical paths in Pipelined processors. Section 7 outlines the proposed architectural overview. Section 8 talks more in detail about the proposed architectural components. Section 9 presents the full system architecture for Single-stage error masking. Section 10 presents full system architecture for Multi-stage error masking. Section 11 highlights the EDA tools used for evaluating results and the Process Design Kit used for simulations. Section 12 proposes the design schematic. Section 13 covers the design simulation. Section 14 is a final comparative analysis with previous existing architectures.

2. PRIOR WORK

Pre-proposed measures available for timing error resilience seek to provide the requisite hardware support for increasing the efficacy of timing errors. Our literature proposes a model of timing error resilience in a sequential manner, categorized into 3 primary domains: Error detection, Error prediction and Error masking.

2.1. Error Detection

Blunder identification strategies square measure uphold recognition data way flags for changes approaching when the clock edge. In [5], perhaps the most punctual circuit for on-line fleeting request mistake discovery abuse a web solidness checker that observed late advances approaching in a strength checking sum when the clock edge was portrayed. [6] and [7], a detecting circuit for defer flaws for self-checking applications was depicted. In [8], blunder location upheld resampling data way flags when a postponement, so correlation the resampled worth to the value hang on inside the data way flipflop was anticipated. RAZOR extended the applying of this on-line worldly request blunder identification subject to downsize force or increment execution abuse runtime voltage/recurrence normalization.

No. [16] proposes a slight distinction of RAZOR, wherein a lock is made to replace the data way flipflop in efforts to reduce irregularities in consistency. Accommodation of the clock obligation pattern is rather imperative to ensure distance from the hold-time requirements conferred by the hook. In [9], a blunder discovering circuit upheld a way electronic gear which will identify every transient game plan mistakes and delicate mistakes was outline. Rationale based strategies for matching postpone testing (upheld circuit duplication have moreover been anticipated. in spite of the fact that inside and out these styles, the dynamic changeability worldly game plan edge is recuperated completely, anyway the blunder is identified once the condition of the framework has been ruined. As a consequence, a rollback and/or a territorial guidance replay leads way for the mistake adjustment overhead.
2.2. **Error Prediction**
Blunder expectation methods square measure upheld watching data way flags for advances for a period defined much prior to the clock edge signal. In [10], a strength checker style forecasts a worldly request mistakes on account of a continuous expansion in postpone because of wear-out and maturing impacts was spoken to. Another mistake forecast strategy that cushions the data way with a postpone part and tests the deferred information way signal in another flip-flop, known as the canary flip-flop, was spoken to in [11]. Expectations of a transient request mistake is due as soon as contrast is observed between the values of a data way flipflop and the insides of a canary flipflop.

Blunder forecast upheld copying basic ways and abuse worldly request mistakes on the copied approaches to anticipate a transient request mistake on the main ways was spoken to in [12]. This methodology is prohibited in its adequacy since 1) the copied and huge ways inside the style could skill totally various remaining tasks at hand and inconstancy and 2) the basic ways could correction over the long run [21]. Also, however the condition of the framework is frequently right, the dynamic fluctuation worldly request edge can't be recuperated because of it's important to include a mistake expectation watch band before the clock edge.

2.3. **Error Masking**
Mistake veiling strategies extended inside the writing might be ordered into 2 classes: coherent and fleeting. Consistent mistake concealing methods (e.g., [13]) utilize repetitive rationale to encode the correct cost of the yield with a more modest postpone once basic ways square measure worked out. Transient mistake concealing procedures veil blunders by time getting, i.e., by executing a strategy to defer the point in season of the true values of data to the successive stages in the pipeline. In [17], a fleeting blunder concealing strategy upheld slow down the clock for one cycle when analyst work a transient request mistake to address the current state of the underlying framework was expected. An underlying supposition of this technique is that the idleness of blunder combinations arising from variations in flip flop from within the styles is an exceeding better sum than a clock cycle to with progress slow down the clock before the state is debased. Nonetheless, in follow, this might be hard to acknowledge in better styles thanks than 1) a small low pr

3. **DESIGN CHALLENGES**

3.1. **Increase in dynamic variation**
Given the scenario where the threshold voltage value is undermined by the value of the supply voltage, a noteworthy change in the operating frequency is observed. Such can be clearly observed from the figure that follows. As the voltage diminishes in value, even a 5% change in the value of supply voltage translates into an increase in the frequency distribution. In an unlikely scenario in proximity to the threshold voltage, frequency differentials as high as 50% can be forecasted.

3.2. **Subthreshold leakage**
There are two degrading effects of subthreshold leakage of power: Firstly, it will result in exceedingly high-power leakage and secondly, as power leakage increases so will the variations.
An irregularly high percentage of sub threshold leakage power is anticipated as values of leakage power remain more or less the same and the active power succumbs to a cubical decrease in its value.

3.3. Degraded Circuit performance

Value of circuit delay changes inversely with values of supply voltage, consequence of which is a serious degradation in performance. Hence near-threshold operation is only limited to applications requiring less throughput and performance/speed/latency is not the priority.

4. TIMING VIOLATIONS AND ERRORS IN FLIP FLOPS

4.1. Setup and Hold time

Setup Time is the minimum time for which the data should be stable at the input before the active edge of the clock arrives. In other words, it is the minimum amount of time before the clock’s active edge that the data must be stable for it to be latched correctly.

Hold time is defined as the minimum amount of time after the clock’s active edge during which the data must be stable. There is a requisite of time by each sequential element for the data to remain stable after arrival of clock edge in efforts to capture data in a reliable manner.

4.2. Metastability

In a scenario where the transition to signal A happens in close proximity to the active edge of Clock signal C2, an expected repercussion is the occurrence of a setup or hold violation. As a consequence, for an infinite period of time, the output signal B can oscillate. Thus, the output is unpredictable and before the next clock edge of C2 comes, it may or may not settle down to any stable value. This phenomenon is referred to as metastability and it is said that the flop FB entered a metastable state.

During metastability, the circuit may induce high transient currents, as both pull-up and pull-down network may become active. Hence metastability effect can be detected by monitoring the drain current (ID). This is also known as quiescent drain current (IDDQ) test. The system state can be recovered by stalling the clock cycle and instruction replay mechanisms.

4.3. Errors and violations

If stability of data is not guaranteed for a flip flop before the setup time from the active edge of the clock, there exists setup violation at that flip flop. Therefore, if there exist variations in the non-shaded region before the active edge of the clock, it represents a setup violation.

If stability of data is not promised after the hold time (Th) from the active edge of the clock, it results in a hold violation at that flipflop. Therefore, if changes are observed in the non-shaded region after the active clock edge, it represents what is termed as a hold violation.

5. INCENTIVES FOR ERROR RESILIENCE

Our research proposes a replacement technique for the resilience of on-line temporal arrangement errors that masks temporal arrangement errors by borrowing time from consecutive pipeline phases, while not requiring roll-back or instruction replay hardware support. In modern processors, the design is implemented by replacing all the essential strategies with the planned flip-flop and complementary circuit, whereas the house or semiconductor count does not increase significantly. The errors resonating from native dynamic variations, also termed as the temporal arrangement errors are cloaked by time borrowing from successive stages in the pipeline. Errors arising from slow dynamical international variations are handled by the planned vogue. The occurrence of a temporal arrangement error is presumably dependent on the dynamic variability present on consecutive clock cycles that affects the essential strategies. For cases similar to the aforementioned, our propositional vogue paves way for the system to run seamlessly without generating errors for numerous cycles with primary aim
of diminishing temporal arrangement errors. This is done by commencing a quick reduction of clock frequency. The observed performance loss owing to a rapid decrease in clock frequency is almost negligible.

6. LATCHES FOR ERROR MASKING

A latch is quite a workaround for building timing error resilient circuit. A latch is transparent for the whole positive level of clock; hence it is self-resilient to errors by itself. Replacing flip-flops with latches allows the late input signal to propagate to the output even after positive edge thereby masking setup time violations as well as metastability issues at the triggering edge. But the late input signal can propagate, and get delayed more gradually at every stage until it surpasses the positive level, after which the latch gets opaque. This issue can be solved by generating an error interrupt to the processor (when late signal arrives after positive edge) which will alert the processor to slow down the clock frequency and avoid multi-stage timing errors. The error signal can be simply generated by using a XOR-gate to detect transitions at input or compare D and Q pins of the latch during the checking period. The error signals generated by XOR-gates can then be simply be consolidated into a single signal with an OR tree. These techniques have been proposed earlier in designs such as RAZOR II [15], TDTB [16], DSTB [16], etc.

6.1. Critical Paths in Pipelined Processors

In the forthcoming paragraph in accordance to the associated image, we summarize critical path allocation between flipflops in an industrial processor. As a rudimentary indicator, three credit points were assigned as low, medium and high. For each of the credit point, one of four bars correspond as percentages of Flipflops having a terminating path located all the way up till the top 10-40 percent of all critical paths. The top twenty percent of critical paths are those that have a critical path delay within the 20 percent factor. For instance, if we consider having a critical path delay of 1 ns, all paths having a delay greater than or equal to 0.8ns would constitute the top 20percent of all critical paths. We can summarize from the graph that the number of flipflops ending in the top 20 percent of the critical paths is around 20-30 percent of the total number of flipflops in that pipeline phase for any processor. In an effort to install about 25 percent of the most critical flipflops with error detecting blocks, a good probability of stage timing error detection can be speculated.

7. PROPOSED ARCHITECTURAL OVERVIEW

The architecture we propose has an underlying framework which seeks to discover a temporal order error at a point in time where the clock edge is leading and consequently mask it by employing time borrowing mechanisms from successive stages in the pipeline. The primary objective of employing such a scheme is to offset the dynamic variability effects whilst controlling instruction replay. A direct consequence of dynamic variability, i.e., temporal order variation, adversely affects the temporal order margin which we intend to recover, which is capable, the checking amount (TC). The period of this checking amount is fastened throughout style.

During the incidence of the initial timing error, i.e., occurrence of a single-stage timing error, the forthcoming data signal can result in a critical timing violation equal to the checking period (TC). The proposed design is capable of masking this single-stage timing error by borrowing time interval of duration equal to TC. The utilization of checking period is highly dependent on the flip-flop which makes sure that it happens in conformity with delay generation. For instance, if the delay generated is less than TC (say 0.4TC), it will utilize only 0.4TC only and save 0.6TC for the next consecutive error. Hence, it can implement continuous time borrowing like a latch. However, the hold time is only limited to the length of checking period (TC). Now depending upon the next pipeline stage, there are two cases:

- If the next pipeline stage is not a critical path or isn’t affected by variability in dynamics, the impending delay of the arriving data signal will remain as it is and will not experience any
further spike. Consequently, a critical hit on the timing violation is expected to occur which will be equal to the checking period (TC), commonly referred to as the single stage timing error. The preceding stage caused a delay however no delay was observed in the present stage. The design need not to borrow extra time in this case. Hence, it can keep the checking period similar to the previous stage (TC). No further delay due to this stage will get propagated to next-stages.

- If the succeeding pipeline stage is affected by dynamic variability and identifies as a critical path, much like the stage before it, then a severe timing violation comes to play as a result of the late arriving data signal, the value of which is double the control period (2TC). The proposed design allows for masking of this two-stage timing error, making use of extra time gaps of duration equal to TC. Hence the maximum allowable delay in this stage can be as high as 2TC.

The proposed design holds the ability to hide upto k-stage timing errors. However, it is noteworthy that, the design can only borrow a maximum time of T/2 from the next stage (which is the length of the positive clock level). Hence, for a k-stage pipeline, it can only borrow a maximum time of T/2 from last pipeline stage. Thus, the positive half cycle can be distributed into k intervals, with each individual duration TC. Following relationship must be followed when deciding the checking period (TC) and the maximum no. of stages (k) up to which the error will be masked.

The examination period holds authority over regulating the hold-time inhibitions for the proposed design. The presence of padding is predominantly important, especially in the short-term paths to ensure that our delay value surpasses that of the control period and the hold time. However, hold time is significantly reduced as compared to latch-based error resilient circuits, which have a hold time of half the clock cycle (T/2). The effective hold time in our design is only increased by TC as compared to normal hold time constraints.

8. PROPOSED ARCHITECTURAL COMPONENTS

- Error Detecting Flip-Flop (EDFF)
- Checking Clock (CCK) Generator
- Late Detector
- Master Clock (MCK) Generator
- Error Flip Flop

9. FULL SYSTEM ARCHITECTURE FOR SINGLE-STAGE ERROR MASKING

Fig 1 depicts the single stage full system architecture.

![Single Stage Full System Architecture](image)
10. FULL SYSTEM ARCHITECTURE FOR MULTI-STAGE ERROR MASKING

Fig 2 demonstrates the two pipeline stages in architecture for multi stage error masking.

11. DESIGN SCHEMATIC

Below is the schematic capture of the proposed design in Cadence Virtuoso® Analog Design Environment. 2-stage pipeline architecture was modelled with one error-detecting FF for each stage. Fig 3 presents the proposed design schematic.
12. DESIGN SIMULATION

The concept has been simulated using Cadence Spectre® Simulator at 384 mV (near-threshold voltage) and 1 GHz clock frequency. Fig 4 presents the design simulation results for the proposed design schematic.

![Fig 4: Design simulation results](image)

It can be confirmed by comparing the D with Q and D2 with Q2 signals that they follow each other, even if D changes after the positive clock edge. This proves that the proposed design works perfectly to mask single-stage as well as multi-stage timing error. It can be observed that CCK2 is shifted whenever error occurs in previous stage. The master clock becomes transparent whenever a change in D occurs during checking period. The error signal is always latched at negative clock edge. For error free case, edge triggering feature of the flop is retained if necessary hold conditions are met (Thold = TC). Since the hold time constraint is very less, the design is also resilient to spurious transitions and glitches.

13. COMPARITIVE ANALYSIS

Our proposed work and objective most closely relate with TIMBER architecture [14]. The detailed area comparison between the two designs is mentioned below. Table 2 presents the comparative analysis of our proposed design and previous architectures.
Table 2: Comparative analysis with previous architectures

- **Flip-Flop Area**: TIMBER flipflop uses a shadow master latch for double sampling the input data. This has an adverse effect on area, as extra no. of transistors will be required for the shadow latch. Our proposed design doesn’t introduce extra circuitry to the already existing Master-Slave Flip-flop. This basically modifies master clock to mask timing error. Hence the proposed design enjoys a huge area savings equivalent to a latch.

- **Verification Complexity**: EDFF is just an MSFF with modified clock, and a complementary XOR-gate. This XOR-gate can be also implemented as a separate standard cell, since it does not require any internal signals of the MSFF. Hence the proposed design is much easier to implement with almost zero modifications in standard cell of an MSFF. It is also compatible with any MSFF like C2MOS, TG-based etc. Using of standard cell prevents any further physical verification complexity which may be incurred if designs like TIMBER is used.

- **Delay Generation Area**: The Delay Generation for various checking periods doesn’t need to be done explicitly at every stage like in TIMBER. In proposed design the previous checking clock was utilized to generate the new checking clock. Hence the delay generation can be sufficiently produced with almost “k” times less inverters as compared to TIMBER.

- **Clock Control Logic Area**: The clock control logic may seem to take much larger area footprint in our case. But this is only one-time investment. Every stage has only one clock control circuit, and the extra area overhead is dominated by Flip-Flops and Delay Generation circuits. TIMBER may prove area efficient for 2-stages and 2 critical path flip-flops, but any number higher than that its size increases significantly as compared to proposed design. A modern processor may have upto 14 pipeline stages with more than dozens of flip-flops in most critical paths for every stage. In this case, our design proves more efficient.

References

[1] N. Pinckney, D. Blaauw and D. Sylvester, "Low-Power Near-Threshold Design: Techniques to Improve Energy Efficiency near-threshold design has been proposed to increase energy efficiency across a wide range of applications," in IEEE Solid-State Circuits Magazine, vol. 7, no. 2, pp. 49-57, Spring 2015, doi: 10.1109/MSSC.2015.2418151.

[2] A. Bal, J. N. Tiwari, J. Narayan Tripathi and R. Achar, "A Novel Programmable Delay Line for VLSI Systems," 2019 IEEE 23rd Workshop on Signal and Power Integrity (SPI), Chambéry, France, pp. 1-3, doi: 10.1109/SaPIW.2019.8781675.

[3] M. Nejat, B. Alizadeh and A. Afzali-Kusha, "Dynamic Flip-Flop Conversion: A Time-Borrowing Method for Performance Improvement of Low-Power Digital Circuits Prone to Variations," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 11, pp. 2724-2727, Nov. 2015, doi: 10.1109/TVLSI.2014.2366918.

[4] C. Huang, T. Liu and T. Chiueh, "An energy-efficient resilient flip-flop circuit with built-in timing-error detection and correction," VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, 2015, pp. 1-4, doi: 10.1109/VLSI-DAT.2015.7114574.
[5] P. Franco et al., “On-Line Delay Testing of Digital Circuits,” Proc. VLSI Test Symp., pp. 167-173, 1994.
[6] M. Favalli et al., “Sensing Circuit for On-Line Detection of Delay Faults,” IEEE Trans. VLSI Systems, vol. 4, no. 1, pp. 130-133, Mar. 1996.
[7] C. Metra et al., “On-Line Detection of Logic Errors Due to Crosstalk, Delay, and Transient Faults,” Proc. Int’l Test Conf., pp. 524-533, 1998.
[8] M. Nicolaidis, “Time Redundancy Based Soft Error Tolerance to Rescue Nanometer Technologies,” Proc. VLSI Test Symp., pp. 8694, 1999.
[9] Y. Tsiatouhas et al., “A Sense Amplifier Based Circuit for Concurrent Detection of Soft and Timing Errors in CMOS ICs,” Proc. Int’l On-Line Testing Symp., pp. 12-16, 2003.
[10] M. Agarwal et al., “Circuit Failure Prediction and Its Application to Transistor Aging,” Proc. VLSI Test Symp., pp. 277-286, 2007.
[11] T. Sato et al., “A Simple Flip-Flop Circuit for Typical-Case Designs for DFM,” Proc. Int’l Symp. Quality Electronic Design, pp. 539-544, 2007.
[12] K. Bowman et al., “Circuit Techniques for Dynamic Variation Tolerance,” Proc. Design Automation Conf., pp. 4-7, 2009.
[13] M.R. Choudhury et al., “Masking Timing Errors on Speed-Paths in Logic Circuits,” Proc. Design Automation and Test in Europe, pp. 8792, 2009.
[14] M. R. Choudhury, V. Chandra, R. C. Aitken and K. Mohanram, “Time-Borrowing Circuit Designs and Hardware Prototyping for Timing Error Resilience,” in IEEE Transactions on Computers, vol. 63, no. 2, pp. 497-509, Feb. 2014, doi: 10.1109/TC.2012.190.
[15] D. Ernst et al., “Razor: A Low-Power Pipeline Based on CircuitLevel Timing Speculation,” Proc. Int’l Symp. Microarchitecture, pp. 7-18, 2003.
[16] K. Bowman et al., “Energy-Efficient and Metastability-Immune Timing-Error Detection and Recovery Circuits for Dynamic Variation Tolerance,” Proc. Int’l Conf. Integrated Circuit Design and Technology, pp. 155-158, 2008.