On the facilitation of voltage over-scaling and minimization of timing errors in floating-point multipliers

https://doi.org/10.1109/IOLTS59296.2023.10224887

Published in:

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
Copyright 2023 IEEE.
This work is made available online in accordance with the publisher’s policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Open Access
This research has been made openly available by Queen's academics and its Open Research team. We would love to hear how access to this research benefits you. – Share your feedback with us: http://go.qub.ac.uk/oa-feedback
On the Facilitation of Voltage Over-Scaling and Minimization of Timing Errors in Floating-Point Multipliers

Abstract—Voltage over-scaling (VoS) may be one of the most effective power reduction approaches, however, it makes circuits susceptible to timing failures. Various techniques were proposed to facilitate VoS by detecting and correcting errors and moving away from traditional voltage and timing guardbands. However, such approaches require the addition of extra redundant hardware leading to area and power overheads, especially in case of large timing errors. Recent, complementary approaches tried to redesign the target circuits, however, they were not yet applied on complex pipelined architectures like floating point multiplier which is extensively used in neural networks and other popular applications. In this paper, we develop a low-power pipelined floating-point IEEE-754 compatible multiplier that can operate reliably under voltage over-scaling (VoS). This is achieved by applying a path-shaping approach that helps minimize the number of paths susceptible to timing errors under lower voltages. In addition, our micro-architectural modifications isolate the critical paths only to a single pipeline stage, thus minimizing the error-prone stages under VoS. To showcase the efficacy of our approach, we perform post-place dynamic timing analysis using various benchmarks, indicating our design can lead up to 171% better SNR, 80% less BER and 36% less power under 18% less voltage compared to the baseline multiplier. The applied analysis reveals that our approach can help limit the erroneous outputs of the unit by up to 74% and reduce by 20% the multi-bit error probability, in such a widely used arithmetic unit.

I. INTRODUCTION

The advent of Internet-of-Things and Artificial-Intelligence based applications led to the growing need to execute reliably various neural networks (NN), and signal-processing algorithms on energy-constrained devices. However, such computationally intensive algorithms require many floating point multiplications, which are among the most complex and power hungry operations in modern CPUs [1], [2], thus requiring special attention. Most prior works focused on optimizing the performance of Floating Point (FP) units, but have given less attention to their power consumption, apart from a few recent ones [3]–[8]. Such works tried to optimize some operations like rounding [3], [5] needed in Floating Point Multipliers (FPM) or introduced new FP formats with less number of bits in the mantissa and/or exponent [4], [8]. Such works may have helped to limit the power consumption of FP units, but require extensive adjustment of each workload to the new FP types and have not yet explored aggressive energy saving approaches like Voltage over-Scaling (VoS) [9]. VoS offers quadratic power gains, but it leads to timing errors due to the incurred delay increase that may not allow many critical computational paths to complete within the clock period [10], [11].

Prior studies have shown that is possible to apply VoS by using in-situ error detection and correction registers [12], [13] or by approximating the arithmetic units and redesigning each target signal processing architecture to gain timing slack for mitigating any VoS induced delay increase [14]–[19]. However, most of these studies focused on integer arithmetic with only very few trying to apply VoS and address any resultant reliability issues like timing errors in FP units. Among these, [20] proposed a model to predict VoS-induced timing errors in FP units, but this did not involve any error mitigation and redesign approach of the FPM. Recently, [21] tried to apply VoS to floating point addition [21], but not to FPM, which is an even more complex operation, requiring a deeper pipeline.

In this paper, we focus on the redesign of an IEEE-754 [22] compliant pipelined FPM for facilitating VoS and addressing at low cost any timing errors. Instead of using extra redundant circuits to detect and correct any errors or reducing the precision as prior works, we redesign the FPM’s micro-architecture to facilitate VoS, while minimizing the probability and impact of errors. Our approach shows that VoS is possible to be applied with limited error mitigation costs, once we move away from performance centric design flows, which tend to lead to circuits with many error-prone long latency paths (LLPs) as we highlight later in the paper.

The contributions of this paper are summarized as follows:

- We redesign a conventional pipelined IEEE-754 [22] compliant FPM [23] by applying micro-architectural changes to isolate the critical paths in a single pipeline stage. By doing so, we limit the pipeline stages that may fail under VoS, while constraining the error propagation across stages. To achieve this we have used a Wallace tree for the processing of the mantissa’s partial products, which also help to accelerate their calculation compared to the conventional architecture.

- We carefully shape the path distribution of the redesigned FPM micro-architectue such that the LLPs are significantly reduced and isolated to as few pipeline stages as possible for minimizing the failure probability under VoS.

- We showcase the efficacy of the developed FPM architecture in terms of Bit-Error-Rate (BER), Signal to Noise Ratio (SNR) and power savings under various scaled voltages by performing post-place-and-route power and dynamic timing analysis (DTA) with various workloads. The DTA allow us to measure the timing violations, and estimate the single- and multi-bit errors on the FPM outputs under various voltages, which was never shown before.

- We measure BER, SNR, and power of the proposed and conventional FPM architectures using various workloads including Neural Networks, Computer Graphics, and Data Mining. To enable this we extracted 200K FP multiplication operands from such workloads to help perform a realistic evaluation. Results show that our architecture compared to the baseline achieves up-to 80% less BER, and up-to 171% more SNR, while consuming up-to 36% less power under 18% VoS, compared to the conventional architecture. In addition, in the proposed architecture we observe up-to to 20% the single and multi-bit error probability. The rest of the paper is organized as follows. Section II discusses the background while highlighting unaddressed challenges. Section III describes our proposed approach; Section IV presents the evaluation of our approach, while Section V concludes the paper.

II. BACKGROUND AND MOTIVATION

In this study, we focus on the consideration of VoS on FPM, thereby we first discuss the basics of such a complex operation and a typical pipelined micro-architecture and then discuss the VoS related challenges.

A. Floating Point Multiplication Architecture

The floating-point representation of a real number \( x \) in the IEEE 754-2008 [22] standard is given by:

\[
x = (-1)^{s} \cdot M_{x} \cdot 2^{E}
\]  

(1)
A floating point multiplication between two numbers $A$ and $B$ can be expressed as:

$$A \cdot B = S_A \oplus S_B \cdot M_A \cdot M_B \cdot 2^{E_A + E_B + bias} \quad (2)$$

In the case of single-precision 32-bit IEEE representation, $S$ has 1-bit length, $E$ has 8-bit length, and $M$ has 23-bit length.

The IEEE 754 standard specifies the floating-point representation of real numbers, where the multiplication of two numbers $A$ and $B$ is performed by multiplying their mantissa and adding their exponents, as shown in Equation (2).

The floating-point multiplier is commonly implemented using pipelining, which involves multiple computation steps, including exponent analysis, normalization, mantissa multiplication, and rounding. A state-of-the-art micro-architecture for a pipelined IEEE-754 compatible FPM is presented in Figure 1, which is part of the latest open-source OpenRISC Morlik Marrochino architecture [23].

The FPM unit is comprised of six pipeline stages, starting with the analysis and pre-normalization of input operands in the first stage. Subsequently, the exponent and mantissa alignment is performed, followed by two pipeline stages dedicated to the mantissa multiplication. The last two stages focus on the rounding operation and combine the final computed result.

As with any pipelined architecture, once traditional design flow tools are applied, each stage of the FPM is composed of a set of $N$ distinct combinational paths $P = p_1, p_2, \ldots, p_N$, each characterized by its respective delay $D(p_i)$ for $i = 1, 2, \ldots, N$. The overall path set can be further distinguished into a set of $K$ Short Latency Paths (SLPs), which can be defined as:

$$P_{SLP} = \{p_{SLP}^1, p_{SLP}^2, \ldots, p_{SLP}^K\} \subset P$$

and a set of $M$ Long Latency Paths (LLPs):

$$P_{LLP} = \{p_{LLP}^1, p_{LLP}^2, \ldots, p_{LLP}^M\} \subset P.$$  

In general, $P_{SLP}$ is excited by a set of operands $O_{SLP}$ and can be completed within a time $T_{SLP} = \max\{D(p_{SLP})\}$, whereas $P_{LLP}$ is activated by a set of operands $O_{LLP}$ and requires more time than $T_{SLP}$: $T_{SLP} < D(p_{LLP}) \leq CP_{STA}$ to be completed. In a pipelined architecture, the longest path delay across all stages determines the clock period (CP), such as:

$$CP = \max_{p \in P}\{D(p)\} \quad (3)$$

### B. Voltage Over-Scaling

Voltage Over Scaling (VoS) is considered the most effective power reduction technique since it can lead to quadratic power savings [9]. However, the reduction of the supply voltage ($V_{dd}$) leads to an increase in the gate delay as given by (4):

$$\tau_d = \gamma \left(\frac{V_{dd}}{V_i}\right)^{\alpha},$$

where $\tau_d$ is the gate delay, $V_{dd}$ is the supply voltage, $V_i$ is the transistor threshold voltage, and $\gamma$ and $\alpha$ are constants that depend on the physical characteristics of the transistor [24], [25].

Consequently, lowering the $V_{dd}$ increases the total delay of any computational path $D(p_i)$ and if it becomes larger than the adopted clock period (i.e. $CP$; Eq. (3) in this case) it will lead to timing errors since the path will not be able to complete. Estimating which paths will fail is becoming even more complicated in pipelined architectures like the FPM that is discussed in Figure 1 since any path $D(p_i)$, especially any LLP within any stage may fail, causing an error which can significantly alter the output. For instance, in case study of the FPM, let’s assume that the error-free output is 350.56, which is represented in the IEEE-754 by the 23-bit mantissa 010111101001110101110 and the exponent 1000111. In case of VoS-induced timing violations, one or more output bits of the mantissa may not be computed correctly, resulting in an erroneous mantissa output 0101111010011101011101110101110, leading to an incorrect number: 478.56. Support for the VoS causes a timing violation and a bit-flip during the exponent calculation or mantissa alignment. In that case, this could result in $1.5\times10^{-12}$ instead of the expected output of 350.58, which can significantly affect the application’s functionality. Therefore, applying VoS on a pipelined design like the FPM is not straightforward. In fact, the manifestation of the timing errors under VoS depends on the processed operands and on which computational paths they excite at every stage and every cycle, which makes it even more difficult to predict, avoid and/or correct timing errors [26].

### C. State-of-the-Art and Challenges

Conventionally, in-situ error detection and correction (EDAC) approaches were proposed to allow VoS and deal with any timing errors [27], [28], [26]. Such schemes integrate special units/registers to monitor the critical LLPs and either change the cycle time/clock frequency [27] or utilize different recovery mechanisms [26], [28] (e.g. pipeline stalling, instruction replay) in case of detected errors. Although such schemes showed that it is possible to apply VoS and deal with timing errors, they make it more difficult to meet the design constraints and may lead to large recovery overheads, especially if the activation probability of the error-prone LLPs is high [26].

Interestingly, the above VoS aware schemes were mainly applied to integer arithmetic rather than FP units. Among the few works that explored VoS on FPM, [29] proposed a low-power probabilistic floating point unit that applied truncation on less significant bits (LSBs) during the mantissa calculation. The probabilistic nature, however, makes it difficult to predict the power-quality trade-offs in that unit and requires careful investigation for each workload. VoS induced timing error models were proposed in [20], [30] and were used in FP units to try to predict critical failing bits and avoid them. A recent approach tried to reduce the error-prone LLPs and applied dynamic bit truncation [31]. However, it was only applied on FP addition and not applied to FPM, which is a more complex operation and requires a deeper, more complex pipelined micro-architecture.

Challenges. To better understand what is needed for facilitating VoS and why existing approaches based on error detection and correction are difficult to be integrated within every register in pipelined architectures we need to consider current EDA design flows. In fact, traditional EDA tools do not favour VoS since they result in path profiles such as the one shown in Figure 2 (a), where many LLPs are close to the targeted clock edge. Such a phenomenon known as timing wall [32] is a consequence of the way modern performance centric EDA tools optimize the architecture subject to a global frequency constraint. Specifically, the LLPs are optimized...
by gate up-sizing, while the many SLPs which are inherently fast are allowed to become near-critical to recover area or power costs. Unfortunately, a path profile such as the one depicted in Fig. 2 (a) has an increased failure probability under any delay increase, since many paths are critical, with low or no timing slack from the clock signal edge. Therefore, designs with such path profiles do not favour VoS. This becomes even more challenging in pipelined architectures, where if every stage has a profile as in Fig. 2 (a), then any of the stages may fail, further increasing the failure probability of the design. Moreover, the existing error detection and correction schemes will need to be used extensively in several stages leading to power and performance penalties, especially in case of many errors [13].

III. PROPOSED APPROACH

The main goal of this work is to address the challenges mentioned in the previous section and redesign a pipelined FPM such that it is less susceptible to timing errors under VoS. This is achieved by applying (a) path shaping, a technique that helps to modify the path profile of the circuit with much less critical paths as indicated in Fig. 2 (b) and (b) critical path isolation that helps to isolate the critical paths in a single stage of the pipeline. By isolating the critical paths to as few pipeline stages as possible we minimize the error-prone locations per operation and the timing-error propagation across these stages at a low cost without additional circuit overheads. In the next paragraphs, we discuss the application of each step of the workflow depicted in Figure 3 in the target FPM architecture.

A. Path Shaping

The goal of this phase is to avoid the timing wall phenomenon and reduce the number of long-latency paths while ensuring all other paths are fast enough to tolerate voltage-induced timing failures. To achieve this, we introduce path-group constraints during synthesis in the Synopsys Design Constraints (SDC) file. By applying multiple path-group constraints, we can avoid optimizations that make naturally fast paths slower for saving area/power. These constraints may impact the area and power consumption, but the resulting overheads that depend on the targeted path distribution can be kept small. Initially, we apply strict constraints to shift the timing wall away from the target CP STA. If there are timing violations after synthesis, we relax the design constraints and re-run the method until the timing target is met. The synthesis step is followed by the place and route using the Innovus Tool from Cadence. Sign-off STA is then performed using Synopsys PrimeTime to verify that the design has achieved the timing closures. To reduce the number of long-latency paths further, we utilize tool command language (tcl) scripts to sort the paths into two different sets, the $P_{SLP}$ and the $P_{LLP}$ based on the worst-case timing within the resulted groups. This automated procedure decreases the number of LLPs while ensuring all other paths are fast enough to tolerate variation-induced timing failures.

B. Micro-Architecture Modifications

The second goal of our design approach is to minimize the number of the few critical LLPs by restricting them to a single stage. This reduces the likelihood of errors occurring at these locations, making their mitigation easier and minimizing the propagation of errors from one stage to another. To achieve this, we modified the microarchitecture by re-distributing operations on different stages, while replacing the original array-based mantissa multiplication with a faster one. In particular, in the initial design, as depicted in Figure 1, the mantissa multiplication operation was performed in a single stage, the fourth one. In the baseline design, in the third stage, the mantissas were split and properly formed to be forwarded to the main mantissa multiplication. In our case, the mantissa multiplication is instead shared in the second and third pipeline stages of the component. The 23-bit mantissa multiplication, in our case, is implemented with a Wallace tree multiplier [33] that uses 4:2 compressors as opposed to the conventional architecture to further reduce the required partial product compression stages, which tend to be much faster. In addition, we combined the logic of exponent and mantissa alignment (second stage of baseline design) with the initial tree part of the Wallace multiplier. By doing so, we can reduce the required pipeline stages by one without unevenly overloading any of these stages. Before we made these adjustments to the pipeline, we made a deep investigation of the critical paths and applied the path shaping procedure explained in the previous subsection. By making such changes, we were able to realize the FPM with five instead of six stages, thus reducing the latency of each operation, without increasing the clock delay.

C. Evaluation

The proposed approach relies on the dynamic activation of paths to eliminate voltage-induced failures and minimize quality loss. The analysis phase utilizes post-place gate-level simulation (GLS) supported by ModelSim to characterize the data-dependent path activation. By monitoring the inputs and outputs of all flip-flops in the design and generating a corresponding event log, we can extract instruction-aware Bit Error Rates (BERs) that depend on the dynamic excitation of critical paths triggered by specific operands under potential delay increase levels. To perform full back-annotated simulation, we provide ModelSim with an RTL netlist, a testbench, and a standard delay format (SDF) file. We obtain the RTL netlist and the SDF file after the PNR step. Our profiling tool feeds ModelSim with real floating-point multiplication operands, creating the required testbench. Given that every set of operands under nominal conditions produces an error-free output, we define this error-free simulation output as Dgold. On the other hand, when a set of operands is executed under variations or scaled voltage, we represent its likely erroneous output as Osim. During this analysis phase, by comparing Dgold to Osim, instruction-aware BERs are extracted. Additionally, this tool can extract an essential value change dump (VCD) file for dynamic power analysis, which contains information about the switching activity and value changes that occurred during the simulation for nets and registers. The VCD file is crucial for identifying switching activities and estimating dynamic power consumption. By utilizing this information, we can further optimize the design to reduce dynamic power consumption and ensure optimal performance.

IV. EVALUATION RESULTS

In this section, we present a comparative analysis of three different architectures: (a) the proposed Redesigned Shaped (RS) FPM architecture, where both approaches mentioned in the previous section are applied, (b) the original FPM architecture (referred to as baseline), and (c) a redesigned version of the FPM without the applied path shaping but only with the micro-architectural changes (referred to as Redesigned Non-Shaped or RNS). Our analysis evaluates the
Fig. 4: (a) Customized micro-architecture and pipeline of the proposed FPM design (modifications in orange) and (b) Path Distribution of the Non Shaped (RNS) and Shaped (RS) FPMs.

performance of these architectures in terms of BER, SNR, Erroneous Outputs and power consumption under various degrees of voltage scaling, including nominal, scaled (8% and 15%), and low (23%) voltages.

A. Experimental Setup

In the experimental setup, we used the Synopsis Design Compiler to synthesize the designs and the Cadence Innovus tool with the NanGate 45 nm cell library for placement and routing. To estimate power and BER, we applied the analysis steps mentioned in the previous section and conducted post-place and route analysis DTA using ModelSim with FP operands from a range of applications. To obtain program traces, we utilized a profiling tool [34] on various FP-intensive applications from fields such as Data Mining, Fluid Dynamics, and Computer Graphics. Specifically, we extracted operands from the BT and CG benchmarks of the NAS suite and multiplication instructions from the Gauss and Srad programs. We also created a CNN inference tool to extract single precision FP multiplication operations from kernels of convolutional neural networks (CNNs). In particular, we extracted multiplication operands from the CIFAR-10 and MNIST datasets.

For DTA, we employed post-place gate-level simulation (GLS) as was discussed in Section III-C. Overall, our experimental setup enabled us to accurately evaluate the performance of the proposed FPM architecture and compare it against the baseline FPM architecture and the redesigned one without shaping. Through this rigorous evaluation process, we were able to obtain a comprehensive understanding of the performance of each design under different conditions.

B. Path Profile

Fig. 5 presents the impact of path shaping on the proposed RS architecture and the redesigned RNS architecture, where no path shaping is applied. These two architectures can operate after the signoff analysis at 769 MHz frequency which is equal to a minimum clock $CP = 1.3ns$. It is evident that the proposed RS architecture has much fewer critical paths compared to the redesigned RNS architecture, where no path shaping is applied. Considering the rounding stage (Stages 4 and 5), the critical paths have been fully eliminated in Rounding Stage 5 in comparison to the RNS design, as it is visible in Fig. 5, making the circuit less prone to timing failures. In contrast, the RNS architecture, even though its micro-architecture is redesigned, still suffers from a high number of critical paths, resulting in a lower timing performance compared to the RS architecture. In the third stage of the pipeline, critical paths close to the timing wall in the final accumulation stage of the multiplier, have been reduced by 50% as it is shown in 5 and these paths have been shifted away from the clock edge, requiring less delay to be executed. Reducing the number of critical paths in this stage is crucial since the mantissa multiplication and final product accumulation are considered to be the most time-consuming operation and vulnerable under VoS in this architecture.

It is worth noting that the RNS architecture without path shaping completes a multiplication in 5 cycles rather than in 6 cycles required.

Fig. 5: Redesigned Non-Path Shaped RNS (Blue) and proposed Path Shaped RS (Red) path distributions across the five pipeline stages
by the baseline architecture. Additionally, it achieves a much better clock delay of 1.3ns, which is significantly lower than the 2ns required by the unmodified baseline design. Note that in the following analysis, we applied the degrees of VoS while executing each design at the maximum frequency (1.3ns in RS/RNS and 2ns in the baseline). As it will become evident also from the following results the proposed RS design does not only minimize the errors under VoS but also runs at higher frequency/low clock period (i.e. 1.3ns instead of 2ns of the baseline) and lower latency (i.e. 5 stages) compared to 6 stages of the baseline.

C. Evaluation of BER and Quality

1) Bit Error Rate: To assess the effectiveness of our approach in eliminating timing failures, we estimated the output bit error rate (BER) of our proposed RS architecture, the redesigned RNS architecture, and the original baseline architecture under 23% voltage configuration. The BER varies across different benchmarks since it is dependent on the dynamic excitation of different paths by the input operands. Figure 6 presents the BER in the sign (bit 32), exponent (31 down to 24), and exponent (23 down to 1) bits of the different architectures. We estimated the BER by comparing each bit of the simulated output $O_{sim}$ for specific operands to the error-free output $D_{gold}$ of the multiplier. The errors mainly arose from Stages 3, 4 and 5 of the pipeline, which perform mantissa multiplication and rounding, as shown in Figure 5. As illustrated in Fig. 6 the initial baseline design as described in Section II, without any micro-architectural changes or path shaping exhibits a significantly high BER when operated under VoS. Especially in the case of Gauss and Srad benchmarks, the BER is at least 10% higher for every bit position, reaching up-to 50% higher in mantissa and up-to 80% in the exponent bits compared to the proposed RS architecture. In contrast, the proposed RS design exhibited a lower bit error rate (BER) across the bit locations and especially in the exponent, than the baseline design, as evident from Figure 6. Analysis of the redesigned component’s path profile in the previous section revealed that the RS configuration (red line) eliminates timing failures caused by the rounding stage in the least significant bits (LSBs). On the other hand, the RNS design without path shaping, despite being redesigned with the same micro-architectural modifications, still had a high percentage of failing bits in the LSBs (blue line), ranging from 5% to 12%, thereby leading to quality loss. This indicates the effectiveness of path shaping which can further help to limit the timing errors in addition to the micro-architectural changes.

As an additional insight, Figure 6 shows that both the modified RS and RNS designs have an increased BER only in bits 8-15 of the mantissa while eliminating completely the failures in exponent bits which are even more critical for output quality. These bit failures in RS and RNS designs are attributed only to stage 3 (rather than in multiple stages as in the baseline) of the pipeline, which performs the mantissa multiplication.

2) Output Quality: Following the BER analysis, to quantify the impact of timing failures on output quality, we estimate the Signal-to-noise ratio (SNR) of our design and compare it with those of the baseline multiplier under the considered applications.

To calculate the SNR, we consider the error-free output of the simulation under nominal voltage $D_{gold}$ as a noise-free signal and we compare it with the outputs $O_{sim}$ derived from our DTA simulations so that the differences in the outputs incurred by VoS-induced timing failures can be accounted as noise. Figure 7 presents the estimated percentage difference in SNR between our proposed RS design, the redesigned RNS design, and the baseline approach. The blue bars correspond to the SNR performance of the RS design, while the red bars indicate the RNS design’s performance relative to the baseline. Notably, the 23% configuration is not included in this figure due to the baseline design’s outputs crashing and producing vastly dissimilar results, leading to near-zero or negative SNR values. This behaviour was first identified in the BER analysis conducted in the preceding subsection, which demonstrated a considerably higher incidence of failures in this component under this voltage configuration. Therefore, given the proposed design’s ability to maintain an acceptable output quality under this configuration, any comparison between its SNR values and those of the baseline would be futile and thus are not depicted in the figure.

Figure 7 demonstrates the superior SNR performance of the proposed RS design compared to the baseline under 8% configuration, with up to 97% improvement for the CIFAR AR benchmark. This difference is significantly greater than the baseline, being greater by at least 10%. Additionally, the RNS design’s SNR performance (first six red bars) is notably better than the same design without shaping (following six red bars), indicating the effectiveness of the path-shaping technique. These metrics provide clear evidence of the efficacy of path shaping in improving output quality. For the 15% configuration, a higher SNR value of up-to 171% is measured for the Srad benchmark. Overall, the percentage increase in SNR is higher for all benchmarks in this voltage reduction level compared to 8%, which can be attributed to the increased failures across the exponent and mantissa bits in the baseline design under scaled voltages. Specifically, the SNR of the baseline design sharply decreases, while the RS and RNS designs are still capable of maintaining the desired output quality. When voltage was aggressively scaled to 23%, it...
was observed that the proposed RS design maintained high SNR, while the baseline design crashed and produced erroneous outputs. Specifically, when compared to the SNR levels of 15%, the RS design had an average SNR decrease of 21.7%, while it maintained an output SNR higher than 38dB which can be considered acceptable for most considered applications.

Finally, we notice that the SNR varies from workload to workload since each benchmark has different data, which excite the critical paths differently. More specifically, we observe that the NNS workloads exhibit a lower BER up-to 7% compared to the other applications in the proposed design in Figure 6 due to the similarity and small range of their data. In addition, we integrated the RS and RNS multipliers in CNN inference for the considered datasets and we observed that the RS design exhibited a 0.2% accuracy degradation in comparison to 1.8% of the RNS.

3) Erroneous Outputs: Furthermore, we performed a comparative analysis of the number of erroneous outputs and the percentages of single and multi-bit failures at different voltage reduction levels. For our evaluation, we classified any output of our component that had at least a 1-bit difference from the expected result as erroneous. In Table I, we showcase the percentage difference BaselineErrorRate – RSErrorRate (%) between the erroneous outputs of the baseline design and the proposed RS design across the considered workloads.

Table I: Percentage (%) difference between the erroneous outputs of the proposed (RS) and the Baseline multiplier design

<table>
<thead>
<tr>
<th>Gauss</th>
<th>Sobel</th>
<th>MG</th>
<th>Scrd</th>
<th>BT</th>
<th>CG</th>
<th>MNIST</th>
<th>CIFAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>8%</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>15%</td>
<td>13</td>
<td>15</td>
<td>28</td>
<td>85</td>
<td>15</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>23%</td>
<td>71.8</td>
<td>62</td>
<td>62</td>
<td>74</td>
<td>45.3</td>
<td>40.1</td>
<td>17</td>
</tr>
</tbody>
</table>

As shown in this table, the Redesigned Shaped (RS) design exhibited a lower output error rate in every application tested under the examined voltage reduction levels. Notably, in the case of Gauss, when operated at 23%, the output was almost completely corrupted (99.9% different outputs) for the baseline design, while the RS design had only 28% erroneous outputs. The same behaviour was observed for the baseline design under 15 and 23% for other applications too. In the reduction level of 23%, the baseline design failed to produce correct outputs while the proposed design maintained an output accuracy as it was depicted in previous subsections during the discussion of the SNR and BER results. It is worth noticing that even under the 8% voltage reduction level, the baseline design had up-to 10% more erroneous outputs that can be critical in most applications.

Additionally, Figure 8 displays the failure probability of a specific number of bits by determining the exact count of erroneous bits in each operand. The depicted figure presents an average failure probability across all benchmarks for the three designs analyzed, and we aggregated the probabilities of more than 6-bits failing simultaneously since this probability was low for our proposed design. The results depicted in the figure reveal two key findings. Firstly, both the RS and RNS designs exhibited a remarkably low probability (0.3 and 1% respectively) of more than 6-bit failure at one output, whereas the baseline design showed a probability of nearly 20%. Secondly, the RS design demonstrated significantly lower single and multi-bit failure rates, consistent with the findings of the BER analysis.

Overall, in terms of reducing the number of erroneous outputs, the RNS approach showcased higher reliability under every voltage reduction level. The results suggest that the RS design has the potential to improve the reliability and efficiency of digital circuits, particularly when operating under reduced voltage conditions, and could find applications in a wide range of low-power systems.

D. Power and Area Analysis

Using the Cadence tools (Voltus and Innovus), we perform dynamic power analysis and measure the area/power overheads of the proposed RS design, the modified but not shaped RNS design and the original one under different voltages. The average power results for all considered operands of the various applications for the different architectures are depicted in Fig. 9. We observe that the proposed RS design achieves up to 36% power reduction under 23% while maintaining higher output quality, as shown earlier. The non-shaped design achieves 33% gains but, as we mentioned, leads to much lower output quality. As depicted in 9, the redesigned and path-shaped five stages design consume on average 8% less power than the baseline one. Note that the path shaping does not come with any area and power overhead once we compare the RS with RNS designs. We also observe that the area overhead of the path-shaped design is only 1% compared to the non-shaped RNS design.

As shown in this table, the Redesigned Shaped (RS) design exhibited a lower output error rate in every application tested under the examined voltage reduction levels. Notably, in the case of Gauss, when operated at 23%, the output was almost completely corrupted (99.9% different outputs) for the baseline design, while the RS design had only 28% erroneous outputs. The same behaviour was observed for the baseline design under 15 and 23% for other applications too. In the reduction level of 23%, the baseline design failed to produce correct outputs while the proposed design maintained an output accuracy as it was depicted in previous subsections during the discussion of the SNR and BER results. It is worth noticing that even under the 8% voltage reduction level, the baseline design had up-to 10% more erroneous outputs that can be critical in most applications.

V. CONCLUSION

In this paper, we presented a framework to improve the reliability of circuit designs in the face of voltage scaling, by addressing timing failures in pipelined cores. We demonstrated the effectiveness of our approach using the complex pipeline design of an FPM as a case study. Our framework involved modifying the multiplier’s path distribution to minimize critical paths and isolating its few critical paths in one pipeline stage via effective micro-architectural modifications. The evaluation results, based on various program traces, revealed that the proposed design exhibited up to 80% lower bit error rates, 171% higher signal-to-noise ratios, and produced up to 20% less erroneous outputs while consuming up to 36% less power under 18% scaled voltage. These findings underscore the potential of our framework to enhance the reliability and robustness of electronic systems, which is critical for ensuring circuit functionality and longevity under voltage scaling.
REFERENCES


