Overscaling-friendly Timing Speculation Architectures

John Sartori and Rakesh Kumar
University of Illinois

ABSTRACT

Processors have traditionally been designed for the worst-case, resulting in designs that have high yields, but are expensive in terms of area and power. Better-than-worst-case (BTWC) design approaches based on timing speculation (TS) [1, 2, 3, 4] have recently gained ground as an alternative to traditional designs by allowing processors to be designed for the average case and still maintain high yields.

In this paper, we characterize the behavior of TS-based designs in the face of voltage overscaling [5] (or undervolting). We show that the power benefits of TS due to voltage overscaling are greatly determined by the design of the circuit architecture. The benefits are small if the underlying circuit has a small range of timing paths, as such circuits produce catastrophic failures in the face of voltage overscaling. Benefits may be limited even for circuits with a wide range of timing paths, due to short path and long path constraints imposed by TS techniques like Razor [1, 2] and EDS [4]. In general, TS-based designs are shown to be not very effective in the face of aggressive voltage overscaling.

We propose two techniques to alleviate the limitations of TS architectures. The two techniques – using adaptable skew for TS and decoupling pipeline stages using local asynchrony – are shown to be effective at reducing both the number of uncorrectable errors in the face of voltage overscaling and the power consumption of the TS architecture.

Categories and Subject Descriptors: B.8.m Performance and Reliability:Miscellaneous **General Terms:** Design,Performance,Reliability

1. INTRODUCTION

Processors have traditionally been designed for the worst-case, resulting in designs that have high yields, but are expensive in terms of area and power. Several *better-than-worst-case* (BTWC) designs [1, 5, 6, 4] have been proposed recently that allow processors to be designed for the average case while maintaining high yields. Typical TS architectures [1, 4] operate at BTWC design constraints, while detecting and correcting timing errors due to frequency, temperature, and voltage variations. The overall effect is improved yield for a given power budget.

Other than improving yield for a given power, one benefit that is often associated with TS-based BTWC designs [1, 4, 6] is that they allow deeper voltage scaling than conventional designs. In Razor, for example, it is assumed that a processor can be run at voltages significantly lower than the nominal input voltage. Any timing violation is assumed to be detected and corrected by the Razor latch. Other TS-based architectures [3, 4] employ similar techniques.

In this paper, we carefully examine the behavior of TS-based designs in the face of voltage overscaling [5] (or undervolting). We show that the power benefits of such designs are greatly determined by the circuit architecture. We characterize two different kinds of adder circuits (Kogge-Stone and Ripple Carry) and show that the benefits can be small if the underlying circuit has a small range of delay for timing paths (e.g., for Kogge-Stone), as such circuits produce catastrophic failures in the face of voltage overscaling [7]. We also show that the benefits of TS architectures like Razor and EDS can be severely limited even for circuits with a wide range of timing paths (e.g., Ripple Carry adder) due to short path and long path constraints. In general, TS-based designs are shown to be ineffective in the face of voltage overscaling, demonstrating the need for alternative techniques to take full advantage of power benefits achievable through aggressive voltage scaling.

We propose two preliminary techniques to alleviate the limitations of TS – one that supplements TS-based error detection, like Razor or EDS, with a dynamically adaptable skew between the main latch and the shadow latch setup times, and one that uses locally asynchronous design to increase the range of possible voltage scaling.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GLSVLSI'10, May 16–18, 2010, Providence, Rhode Island, USA.
Copyright 2010 ACM 978-1-4503-0012-4/10/06 ...$10.00.

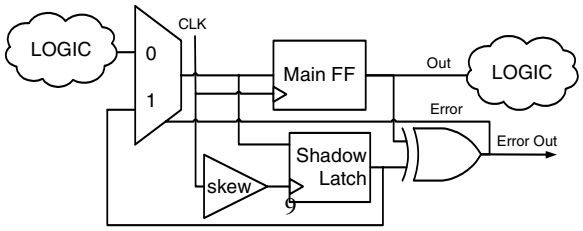

Figure 1: The Razor flip-flop[1].

2. VOLTAGE OVERSCALING LIMITATIONS OF TS ARCHITECTURES

In this section, we demonstrate the limitations of TS designs in the face of voltage scaling. We will consider Razor as a canonical TS design for our discussion.

2.1 Razor Basics

Razor is a circuit-level technique for detecting and correcting timing errors. It detects timing violations by supplementing critical flip-flops with a shadow latch that strobes the output of a logic stage at a fixed delay (which we refer to as *skew*) after the main flip-flop. Thus, if a timing violation does occur, the main flip-flop and shadow latch will have different values, signaling the need for correction. The skew between the main flip-flop and the shadow latch is often chosen to be half a cycle.

Error correction in Razor-based designs involves recovery using the correct value(s) stored in the shadow latch(es). A pipeline restore signal is generated by OR-ing together error signals of individual Razor flip-flops. The signal overwrites the shadow latch data into the errant flip-flop. Other recovery mechanisms for Razor-based designs include the use of clock gating [8] and a counter flow pipeline [9]. The occurrence of metastability at the main flip-flop output is flagged using an additional detector.

Figure 1 shows the Razor flip-flop. More details on the design and operation of Razor can be found in [1].

2.2 Razor Limitations

To guarantee correctness, Razor requires two conditions to be met on the circuit delay behavior - the short path constraint and the long path constraint. The long path constraint (Eqn. 1), states that the maximum delay through a logic stage protected by Razor must be less than the clock period (T) plus the skew between the two

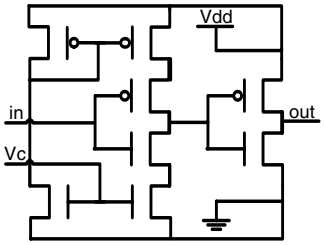

Figure 2: The delay of the current starved delay element increases as control voltage (V_c) decreases.

clocks (clock for the main flip-flop and the clock for the shadow latch).

$$delay_{max} < T + skew \quad (1)$$

If the long path constraint is not satisfied, false negative error detections can occur when a timing violation causes both the main flip-flop and shadow latch to latch the incorrect value. The short path constraint (Eqn. 2) states that there must not be a short path through a logic stage protected by Razor that can cause the output of the logic to change before the shadow latch latches the previous output.

$$delay_{min} > skew + hold \quad (2)$$

Failure to satisfy the short path constraint leads to false positive error detections when the logic output changes in response to new circuit inputs before the shadow latch has sampled the previous output. Combination of the short and long path constraints (Eqn. 4) demonstrates that Razor can only guarantee correctness when the range of possible delays for a circuit output falls within a window of size $T - hold$.

$$skew + hold < delay < T + skew \quad (3)$$

$$delay_{max} - delay_{min} < T - hold \quad (4)$$

Note that Equation 4 implies a tradeoff between the limit of Razor protection and the range of Razor usability. While increasing skew can reduce the number of uncorrectable errors by protecting longer path delays, this also leads to a reduction in the range over which Razor can be applied to gainfully correct errors, since short path violations will increase.

Section 5 characterizes the voltage overscaling limitations of TS-based designs for two canonical circuits. Note that some of the timing constraint violations can be eliminated through *dynamic retiming* [10]. However, short path constraint violations continue to pose a problem.

3. IMPROVING TS EFFECTIVENESS

3.1 Adaptable Correction Window

To address the limitations of Razor imposed by short and long path constraints, we propose an enhancement to Razor to allow an *adaptable correction window*. The enhanced Razor design now incorporates a variable delay element (VDE) [11, 12] that can modify the skew between the main flip-flop setup time and shadow latch setup time based on supply voltage. The VDE is designed such that skew between the main flip-flop setup time and shadow latch setup time increases as supply voltage (connected to V_c) scales down. This behavior can be accomplished by replacing the skew buffer in Figure 1 with a VDE, like the current starved delay element [12] of Figure 2.

To understand the benefits of the enhanced Razor design, note that Razor only provides protection over a limited window of voltage for a given cycle time, as demonstrated by Equation 4. The enhanced Razor design supports an adaptable skew such that the window of correction shifts to the region where it can provide maximum error protection without inducing any false positive error detections. Through transistor sizing, the VDE is tuned to mimic the minimum delay of the protected logic path as voltage is scaled. In this way, Razor's correction window is maximized over the range of input voltages. Note that even if the VDE is tuned conservatively or affected by variation, the adaptable correction window can still provide enhanced error recovery compared to the traditional Razor design, since it amplifies Razor's window of correction at lower voltages, when timing violations are more likely to occur. Another work proposes to adjust the skew of an error detection flip-flop to detect errors at lower voltages [13], however, they use a larger *static* skew that is not adaptable and requires extensive hold buffering.

Specifically, dynamic adaptation of skew affords an extended region of correction when voltage is aggressively scaled, while avoiding false error detections which necessitate costly buffering on paths that do not produce errors. With the adaptable skew technique, we eliminate the problem of Razor induced false positive errors that can make Razor unusable in certain operational regions. This benefit can be translated directly into power savings, since this reduces or eliminates buffering that may otherwise be required to satisfy short path constraints.

Since the delay of the VDE depends on parameters that may vary in fabrication, such as threshold voltage, one practical consideration of note for the adaptable skew technique is that the delay characteristic of the VDE can only be accurately determined post-silicon. Thus, if a finely tuned delay characteristic is needed, the design should incorporate a method for tuning the VDE after fabrication. One such method involves running a known test set through the circuit and tuning the nominal control voltage of the VDE until the point where no errors are observed. Through experimentation, we have observed that post-silicon tuning is likely to be unnecessary in most cases, since a finely tuned delay characteristic is not required to achieve maximum benefit from the adaptable correction window.

An analysis of the benefits of the enhanced Razor design with a VDE is presented in Section 5.

3.2 Locally Asynchronous Design

Another alternative to TS-based better-than-worst-case design is employing local asynchrony (FIFO queues) to decouple timing critical pipeline stages and extend the range of voltage scaling. Figure 3 demonstrates the concept. The enable signal is generated by the decoupled logic to signal that computation has finished for the current inputs. Setting the signal enables writing the current output to the output queue and reading the next input from the input queue. One way to generate the enable signal (which we use in our tests) is to monitor transitions on critical signals that indicate completion of an operation. In the case of the adder, these critical signals are the carry bits. Following the last transition on a carry bit, the addition operation completes after a fixed delay. Figure 4 demonstrates this procedure for an adder circuit. The top input to the XOR gate in Figure 4 represents the path from carry-in to carry out in a full adder, plus a delay margin. Carry transitions launch transitions down this path, and the output of the XOR gate remains high as long as transitions continue. When transitions on all carry bits have ceased, all inputs to the NOR are low, and the enable bit is driven high. One disadvantage of using transition detect logic to generate the enable signal is that critical signal identification is circuit-dependent. Another approach to completion determination is current monitoring. The idea behind this approach is that a decoupled logic stage only consumes dynamic power during computation. Once the computation finishes, the current draw drops to the static leakage current. Thus, an on-die current sensor [14, 15] can be used to monitor the current drawn by the logic and set the enable bit when the current falls below the threshold between operating current and leakage current. The current monitoring approach does not require identification of circuit-specific critical signals but instead requires the designer to assign a bound on leakage current for a circuit block. A conservative threshold can tolerate more variation but may result in reduced throughput.

There exists a tradeoff between the cost of completion determination logic and latency with the locally asynchronous design approach. Performing the determination at finer granularity increases the cost of detection logic in terms of area and power consumption.

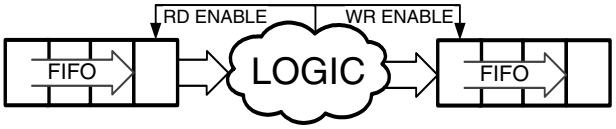

Figure 3: FIFOs are used to decouple a pipeline stage. The logic generates an enable signal which communicates externally that computation has finished for the current inputs.

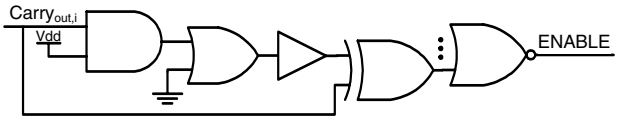

Figure 4: Transitions on carry bits in the adder launch transitions down the modified carry path, which represents the time for addition to finish after the last carry transition. Once the last carry transition occurs, all inputs to the NOR gate go to zero, setting the enable bit.

However, this reduces the uncertainty in determining when the result is complete for the current inputs, thus reducing the latency of the operation. For the adders, we incorporate completion detemrmination logic in each bit. Thus, our power and performance data assume the maximum power overhead and minimum performance degradation. The pipeline stalls or overflows when a queue is full. Pipeline overflows cause timing violations.

Note that the proposed approach has similarities to Synchronous Elastic Flow (SELF) [16]. However, unlike SELF, which provides a model *protocol* for a fully asynchronous design, we provide a specific locally asynchronous *implementation* to alleviate timing violations that occur in traditional designs in the face of voltage overscaling.

4. METHODOLOGY

4.1 Simulated Circuits

To investigate the limitations of TS for a wide spectrum of circuits, we selected two representative architectures that exhibit opposite timing behaviors and represent large classes of circuits.

The first circuit that we use to characterize the limitations of TS-based designs is the Kogge-Stone adder (KSA). Note that many other paths in the KSA architecture have lengths close to the longest path. Therefore, the KSA may exhibit a critical operating point [7] (confirmed in Section 5) akin to traditional high performance processor designs. Characterizing the effectiveness of TS for a Kogge-Stone adder provides a good representation of the effectiveness of TS in the face of voltage overscaling for traditional high performance processor designs.

The second circuit that we used to evaluate the effectiveness of TS in the face of voltage overscaling is the ripple-carry adder (RCA). The RCA architecture consists of timing paths whose delays depend on the length of the carry chain. So, while the path corresponding to the LSB has the least delay, the path corresponding to the MSB has the longest potential delay. Timing violations for such designs are strongly input dependent and may not be massive (confirmed in Section 5) in the face of undervolting. TS may, therefore, be more effective for such designs.

Smooth gradation in path lengths has recently been advocated for high performance processor designs in the context of stochastic processor design [17, 18, 6, 19]. So, our evaluations using the RCA also estimate the effectiveness of TS-based techniques for such processor designs.

4.2 Simulation Details

For circuit characterization, we implement the ripple carry and Kogge-Stone adders using IBM9SF 90nm CMOS FET technology. Each adder architecture is then optimized for circuit performance (speed) regardless of the power consumption and circuit area. This is more realistic in practice than optimizing all the adders to operate at the same clock frequency. Adder circuits are modeled in HSPICE, and exhaustive simulations are run to characterize circuit path delays for different supply voltages. Delay data is then used to annotate RTL descriptions of adder architectures, and RTL simulations are run in Cadence to characterize dynamic delay behavior. For RTL simulations, the input set is composed of 180K random input samples.

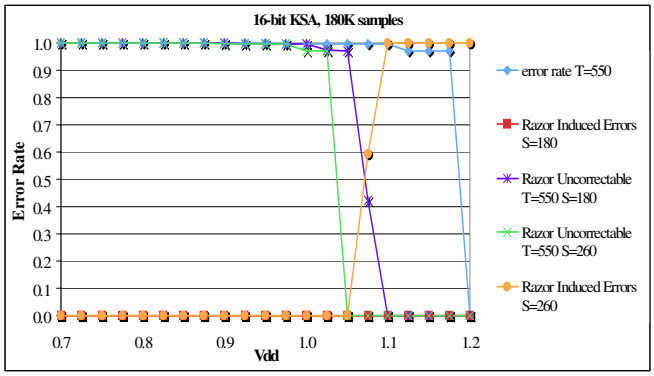

Figure 5: The Kogge-Stone adder has a critical operating point (1.2V for these design parameters). When voltage is scaled below the critical operating point, catastrophic failure occurs. Razor can correct errors over some range, but the extent of scaling is limited due to the critical wall characteristic of the circuit. Increasing clock skew between the clocks of the main and shadow latches actually decreases the range of Razor correction, since it makes Razor unusable at higher voltages without extending Razor's useful range equivalently into the lower voltage range.

Timing results from RTL simulations are processed to determine various error characteristics of circuits under test. For example, to determine error rate at a particular voltage, we simulate for a long clock period and measure the time required for an operation to produce a stable, correct result at the circuit output. This time is compared to the testing clock period to determine when a timing violation has occurred. Power consumption for the circuits is reported by Synopsys PrimeTime.

5. RESULTS

Figure 5 shows the effect of voltage overscaling on the reliability of a Kogge-Stone adder. Reliability is measured in terms of the percentage of the 180K samples that resulted in incorrect outputs (error rate). The results are shown for a Kogge-Stone adder without Razor support (*error rate T=550*) and with Razor support (*Razor Uncorrectable*). Errors occur for Razor if the conditions outlined in Section 2.2 are not met. Results are also shown for different values of skew (S) between the clocks for the main latch and the shadow latch.

There are several things to note in Figure 5. First, the Kogge-Stone adder is indeed representative of the time delay distribution of high performance processors, as it demonstrates critical operating point behavior. As shown in the error curve of Figure 5, scaling beyond a certain voltage leads to a catastrophic failure of the adder (i.e., 100% error rate). Aggressive voltage scaling, therefore, is not possible for such designs.

Second, Razor can provide error correction only over a limited voltage region for KSA, represented by zero uncorrectable errors. This is because in all other regions there are uncorrectable errors due to violation of long path constraints. Even in the region that has zero uncorrectable errors, the power consumption actually increases drastically in spite of voltage scaling. This is because the absolute error rate is high (close to 100%) and the overhead of error recovery for Razor is roughly an order of magnitude more expensive than the cost of performing a normal addition [1]. So, for designs like KSA where timing paths are bunched up (like in traditional high performance processor designs), Razor may not be very effective in terms of power reduction through voltage overscaling (i.e., scaling beyond the voltage for which the first timing violation

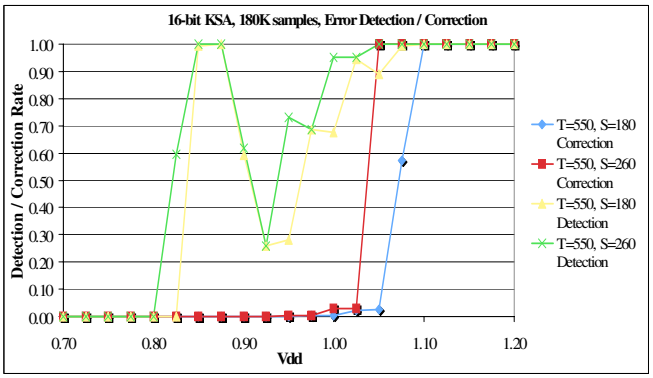

Figure 6: **Error detection and correction for the KSA.**

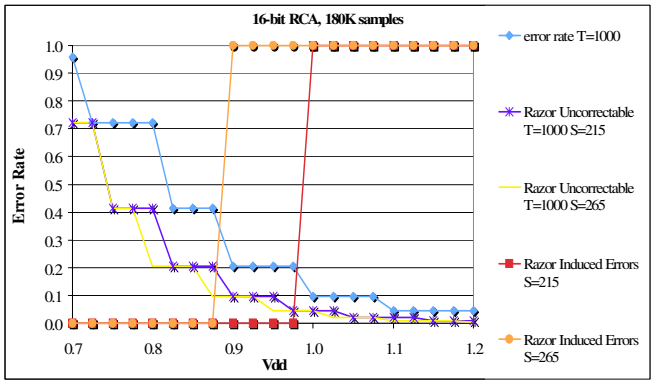

Figure 7: **The ripple carry adder exhibits a wide range of delay paths, characteristic of a circuit that fails gracefully. Error rate for the circuit increases as supply voltage decreases.**

appears). While some power can be saved by eliminating the voltage guardband, scaling past the critical operating point results in nearly 100% erroneous computations.

Another thing to note in the figure is that it is not always a good idea to keep Razor turned on. This is because of potential short path constraint violations, especially for large skews. Failure to satisfy the short path constraint leads to false positive error detections (*Razor Induced Errors*) when the logic output changes in response to new circuit inputs before the shadow latch has sampled the previous output. The practical result for the KSA circuit is that an error is triggered for every operation until the short path constraint is met, making Razor unusable over a range of voltages, as demonstrated by the Razor induced errors in Figure 5. So, the use of Razor in architectures should be optional and determined by the skew, clock period, and input voltage.

Figure 6 breaks error recovery into detection and correction, demonstrating that the range over which Razor can detect errors extends past the range over which Razor can correct errors for designs like the KSA. However, since Razor correction is always on, even when the long path constraint is not met, these extra detections represent wasted power. These facts motivate the need for new design techniques that do not fail catastrophically and error correction techniques that take advantage of the extended window of detection without forcing erroneous corrections.

The ripple carry adder (RCA) architecture is not subject to catastrophic failure in response to scaling past the point of first error. Instead, as Figure 7 demonstrates, error rate increases gradually as voltage decreases. Although the minimum delay for any path of the RCA equals the delay of the sum path of a full adder, operational delay ultimately depends on adder inputs, which generate carry chains from lower to higher order bits. The RCA exhibits maximum delay when the carry chain extends from the least significant bit to the most significant bit. However, on average, carry chains are much shorter, leaving extensive room for aggressive scaling past the point where errors begin to occur. In fact, the error rate reaches close to 100% only at very low voltages.

The above behavior of RCA may be a suitable desired behavior for high performance processor designs to enable significant power savings through voltage overscaling. Recent attempts [17, 18] at processor designs that produce graceful degradation in reliability in the face of voltage scaling try to mimic this behavior.

The error detection rate for the RCA circuit is 100%. This is due to the wide range of delay paths that affect circuit outputs, eliminating the occurrence of false negative errors when the long path constraint is not met. However, correction rates for the RCA can be low in the face of aggressive scaling. These facts demonstrate the error detection advantage of designs that fail gracefully as well as the need for new techniques that can provide enhanced error correction under aggressive scaling.

One may be tempted to conclude from our previous discussion on critical operating point behavior that TS design techniques such as Razor should perform well for architectures that fail gracefully, since such designs do not have a wall of criticality. However, analysis of the results in Figure 7 reveals some serious limitations of using Razor, even in such architectures.

Limitations arise due to the potential short path and long path constraint violations as discussed in Section 2.2. If the long path constraint is not satisfied, erroneous corrections can ensue, and false negative detections can occur if the main flip-flop and shadow latch both latch the incorrect value. In Figure 7, this condition is demonstrated by uncorrectable errors for Razor. Similarly, the failure to meet short path constraints makes Razor unusable over a range of voltages without extensive buffering, as demonstrated by the Razor induced errors in Figure 7.

In fact, the same factor that makes the error behavior of RCA graceful (wide range of path delays) makes Razor less effective. This is because Razor requires the range of delays to be less than a threshold (see Section 2.2). The variation in delay is significantly larger for an RCA design than a KSA design. Figure 8 shows the ranges of possible delay for KSA and RCA architectures at different voltages. In order to make Razor work for circuits that fail gracefully, buffering must be used to increase the delay of short paths, thus shifting them into the window of correction. This buffering adds area and power overheads in a design, negating some of the power savings afforded by better-than-worst-case design. Secondly, required buffering increases the delay on short paths, transforming a circuit from one that fails gracefully to one that fails catastrophically, thus limiting the extent of possible scaling.

So, while TS is ineffective for circuits like KSA because of massive timing violations in the face of voltage overscaling, it is also not very effective for circuits like RCA due to a large span between the maximum and minimum circuit delay. These results demonstrate the inadequacies of current TS-based design methodologies (like Razor) in terms of voltage scaling, motivating the need for new techniques for processor design and error handling.

One technique to enhance the effectiveness of Razor-based TS is to use an adaptable correction window as described in Section 3.1. Figure 10 shows the potential of adaptable correction window to provide added error protection as voltage is aggressively scaled in a circuit that fails gracefully (RCA).

There are two things to note in this graph. First, the adaptable correction window improves the effectiveness of Razor across the entire voltage range. Improvements are both in terms of the percentage of uncorrectable errors as well as the range over which all the errors are correctable. Second, there are no regions now where Razor triggers false positive detections. This is because the VDE is designed such that the skew it introduces always respects the short path constraint. Another benefit attributable to this fact is that the adaptable skew technique eliminates power and area overhead in a Razor design due to required buffering of short paths. However, using variable delay elements does add a constant overhead to a design, since the overhead of each Razor FF increases slightly.

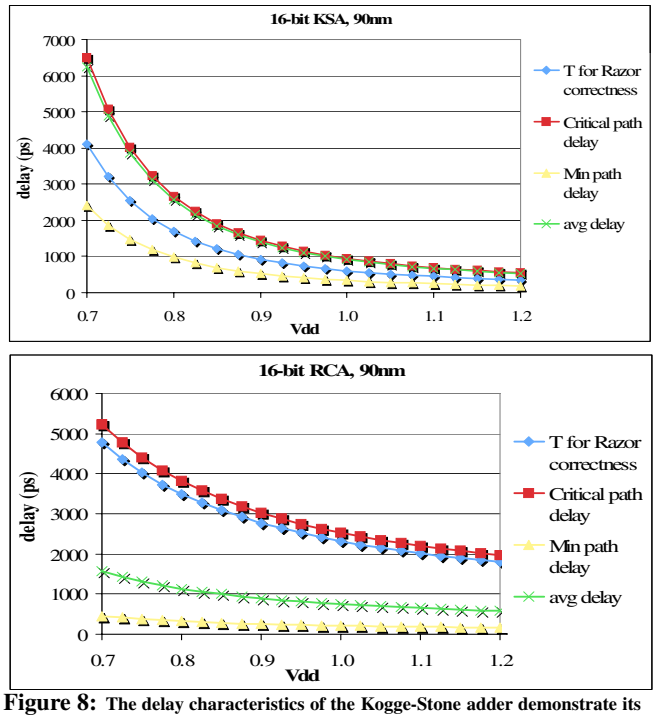

Figure 8: **The delay characteristics of the Kogge-Stone adder demonstrate its critical wall behavior and unsuitability for aggressive scaling. The wide range of delays for the ripple carry adder demonstrate its capacity to fail gracefully, as well as the large margin for power savings with aggressive scaling.**

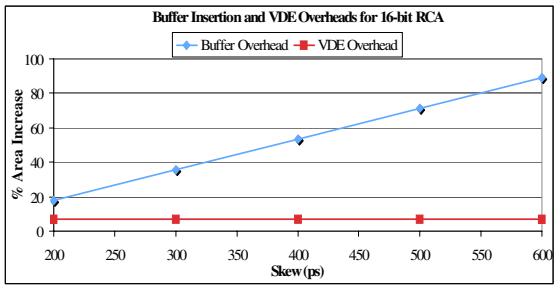

Figure 9: **As skew increases, the number of uncorrectable errors decreases. However, increased error correction comes at the cost of increased buffering overhead, required to make Razor work when short path constraints are not satisfied. With an adaptable correction window, buffering overhead is eliminated and only a constant overhead is introduced for insertions of VDEs.**

Figure 9 compares the overhead of hold buffering against the overhead of VDE insertion for different static skews. Not only does the adaptable skew technique produce lower error rates, but it also has lower implementation overhead.

Lower implementation overhead translates into the ability to correct more errors at a reduced cost. This increased efficiency means that for the same error rate, a design that incorporates adaptable skew consumes less power than a traditional Razor design with static skew. Figure 11 demonstrates the power benefits of adaptable skew design for architectures that fails gracefully (RCA). In addition to traditional Razor, we compared against Razor without buffering to show the power overhead of hold buffering in traditional Razor. Power reduction is an additional 30% compared to the minimum power achieved by the traditional Razor design.

Figure 12 demonstrates the limited effectiveness of adaptable correction window in architectures that fail catastrophically (KSA). While dynamic skew adaptation lengthens the window of correction slightly while eliminating false detections, critical failure still occurs and aggressive scaling is not enabled for traditional architectures. This further motivates the fact that processors need to be

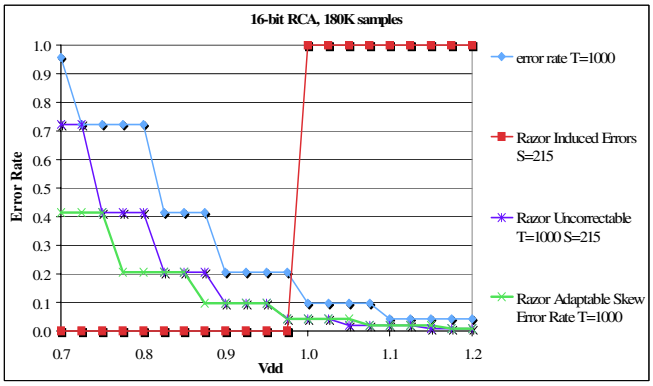

Figure 10: **Adaptable correction window enhances error protection for aggressive scaling, reducing the number of uncorrectable errors. The technique also increases the usable range for Razor by eliminating false error detections.**

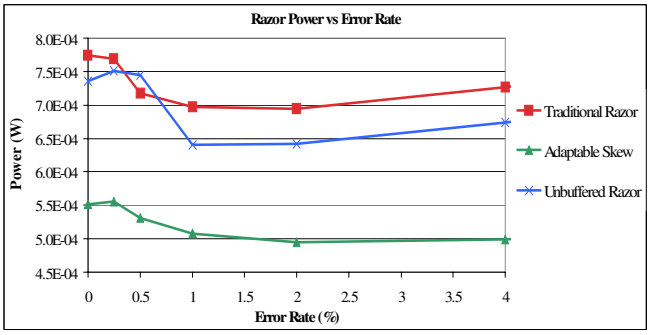

Figure 11: **The adaptable skew technique reduces both implementation cost and uncorrectable errors, resulting in the ability to correct more errors for the same power or achieve the same correction rate at reduced power.**

designed differently (i.e., they should produce graceful timing error degradation).

In a nutshell, Razor has limited effectiveness in the face of voltage overscaling for circuits that demonstrate critical operating point behavior as well as for circuits that have spread time delay distributions. The effectiveness of Razor can be somewhat increased by introducing adaptable skew between the clocks of the main latch and the shadow latch. However, in some scenarios a need remains to follow different design and error handling methodologies to facilitate aggressive voltage scaling and potential for significant power savings.

Pipeline stage decoupling is an alternative technique that attempts to address the limitations of TS-based designs. Figure 8 shows the average and worst-case delays for both the KSA and RCA. While the RCA (gradual failure) shows great potential for LAGS implementation, due to the substantial difference between average and worst case delay, KSA (catastrophic failure) shows little room

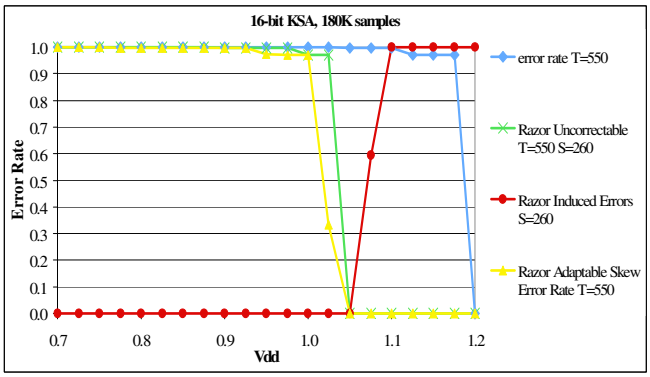

Figure 12: **Adaptable Correction Window for KSA**

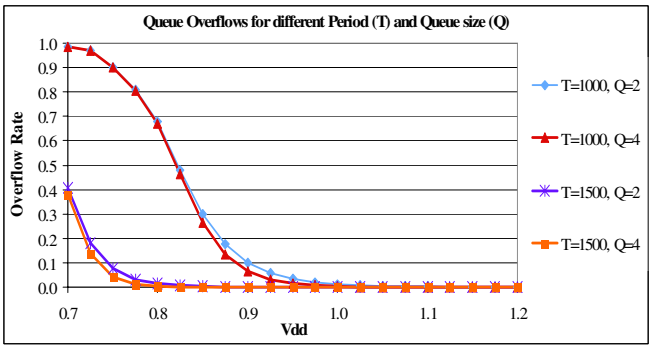

Figure 13: Queue overflows remain low when clock period is less than average latency, but increase steeply afterward. This demonstrates that throughput for the decoupled logic is governed by average latency.

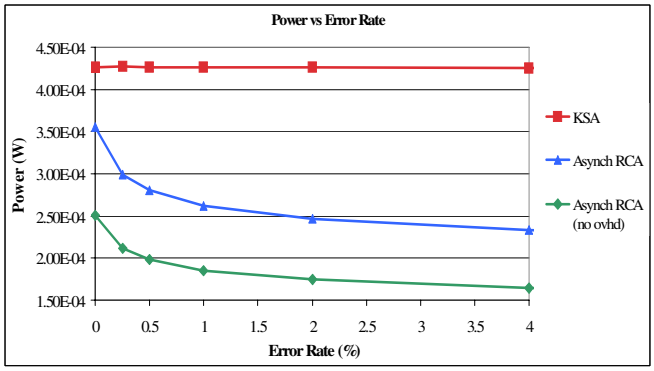

Figure 14: Locally asynchronous design allows the RCA to operate at average latency and keep up with the highly optimized KSA. Even with the overhead of logic decoupling, the decoupled RCA consumes less power than the KSA for the same error rate. Also, reliability can be traded for additional power savings due to the graceful failure characteristic of the decoupled design.

for improvement. The figure also demonstrates another interesting point – RCA performs better than KSA in the average (latency) case, even though the critical path delay of RCA is well known to be substantially higher than that of KSA.

We evaluated pipeline decoupling for circuits with gradual failure characteristics and average latency substantially lower than worst case latency (RCA). When pipeline stalling is used (as in Razor), all timing violations can be avoided. However, stalling pipeline stages may induce unwanted affects on performance. Thus, we consider a case where an overflow of the input queue forces a value past the enable guard, resulting in a possible timing violation. Figure 13 shows how queue overflows vary with respect to voltage for different clock periods and input queue sizes.

Figure 13 demonstrates a few interesting features of pipeline decoupling. First, the shape of the curves indicates that overflows remain very low as long as the clock period is less than the average latency of the circuit, but increase steadily when this condition is not met. This fact suggests that voltage of decoupled stages should be tuned so that average latency equals the clock period, since it is average latency that governs throughput for such a design. In actuality, throughput is slightly less than average latency of the locally asynchronous circuit, since the circuit can only process inputs as fast as they are fed to the input queue. Thus, the decoupled stage can only catch up when it falls behind, but cannot pull ahead when it has processed all available inputs. Another intuition drawn from the figure is that the size of the input queue does not matter much and can be reduced in size to minimize the overhead of this technique. The marginal benefit of increasing the queue size past 2 was limited, even for aggressive clock periods.

To quantify the benefit of logic decoupling for gracefully failing architectures, we created a locally asynchronous version of the RCA and compared against the optimized KSA. Results are shown in Figure 14 (queue size = 2). Although the KSA is much faster than the RCA in the worst case (much shorter critical path), the decoupled RCA is able to operate at the same frequency and voltage with fewer errors (errors for the decoupled design are due to queue overflows). Also, since the KSA is a highly optimized adder architecture, the decoupled RCA consumes less power than KSA, even considering the overhead of queues and completion detection logic. Figure 14 also includes data assuming no overhead for the logic decoupling. This provides a lower bound on the attainable power consumption for this technique, which can be approached with more efficient detection and queueing logic.

6. SUMMARY AND CONCLUSION

Timing speculation architectures like Razor and EDS help improve yield, and are often considered good for power reduction as well, due to reduced voltage margins. In this paper, we examined the effectiveness of voltage overscaling for two TS-based designs. The first design was a Kogge-Stone adder (KSA) that demonstrates critical operating point behavior similar to modern high-performance microprocessors. The other design was a ripple carry adder (RCA) which produces a graceful degradation in reliability in the face of voltage overscaling.

Our experiments showed that TS is ineffective in the face of voltage overscaling for designs in which the timing paths are bunched up, due to massive timing errors upon breaching the critical operating point. The effectiveness of TS architectures like Razor and EDS is limited even for designs with spread path delay distributions. This is because timing variation within a circuit must be less than a threshold for error recovery to work for that circuit. The limitations of TS in the face of aggressive voltage scaling, coupled with the expectation of high variability in coming technology generations, motivate the need for alternative techniques that will allow full extraction of the power benefits available from voltage scaling.

We presented two techniques to alleviate the overscaling limitations of TS. Both *adaptable skew* and *decoupling pipeline stages using local asynchrony* were shown to be effective in reducing uncorrectable errors and power consumption. As power becomes a zero-order design constraint, the value of innovative solutions that make processors more amenable to voltage overscaling will continue to increase.

7. REFERENCES

[1] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *MICRO 36*, 2003.

[2] D. Blaauw, S. Kalaiselvan, K. Lai, W.-H. Ma, S. Pant, C. Tokunaga, S. Das, and D. Bull, "Razor ii: In situ error detection and correction for pvt and ser tolerance," in *ISSCC'08*, 2008.

[3] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *VTS '99: Proceedings of the 1999 17TH IEEE VLSI Test Symposium*. Washington, DC, USA: IEEE Computer Society, 1999, p. 86.

[4] K. Bowman, J. Tschanz, C. Wilkerson, S.-L. Lu, T. Karnik, V. De, and S. Borkar, "Circuit techniques for dynamic variation tolerance," in *DAC '09: Proceedings of the 46th Annual Design Automation Conference*. New York, NY, USA: ACM, 2009, pp. 4–7.

[5] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *ISLPED '99*. ACM, 1999.

[6] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Recovery-driven design: A methodology for power minimization for error tolerant processor modules," in *DAC'10*, 2010.

[7] J. Patel. (2008) Cmos process variations: A critical operation point hypothesis. [Online]. Available: www.stanford.edu/class/ee380/Abstracts/080402-jhpatel.pdf

[8] Q. Wu, M. Pedram, and X. Wu, "Clock-gating and its application to low power design of sequential circuits," vol. 47, 2000, pp. 415–420.

[9] R. F. Sproull, I. E. Sutherland, and C. E. Molnar, "The counterflow pipeline processor architecture," vol. 11, no. 3. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994, pp. 48–59.

[10] S. Lee, S. Das, T. Pham, T. Austin, D. Blaauw, and T. Mudge, "Reducing pipeline energy demands with local dvs and dynamic retiming," in *ISLPED '04*. New York, NY, USA: ACM, 2004.

[11] M. Saint-laurent and H. Samarchi. (2003) Variable delay element. [Online]. Available: http://www.freepatentsonline.com/y2003/0001649.html

[12] G. S. Jovanovic and M. K. Stojcev, "Current starved delay element with symmetric load," *International Journal of Electronics*, 2006.

[13] M. Kurimoto, H. Suzuki, R. Akiyama, T. Yamanaka, H. Ohkuma, H. Takata, and H. Shinohara, "Phase-adjustable error detection flip-flops with 2-stage hold driven optimization and slack based grouping scheme for dynamic voltage scaling," in *DAC '08*. New York, NY, USA: ACM, 2008.

[14] C. Kim, K. Roy, S. Hsu, R. Krishnamurthy, and S. Borkar, "An on-die cmos leakage current sensor for measuring process variation in sub-90nm generations," 2005.

[15] C.-W. Lu, C.-L. Lee, and J.-E. Chen, "A fast and sensitive built-in current sensor for iddq testing," in *IDDQ'96*. Washington, DC, USA: IEEE Computer Society, 1996, p. 56.

[16] J. Cortadella, M. Kishinevsky, and B. Grundmann, "Synthesis of synchronous elastic architectures," in *DAC '06*. New York, NY, USA: ACM, 2006, pp. 657–662.

[17] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in *ASPDAC'10*, 2010.

[18] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Designing processors from the ground up to allow voltage/reliability tradeoffs," in *HPCA'10*, 2010.

[19] R. Kumar, "Stochastic processors," in *NSF Workshop on Science of Power Management*, 2009.

Acknowledgments This work was supported in part by Intel, NSF, GSRC, and the Arnold O. Beckman Research Award. Feedback from Janak Patel, Naresh Shanbhag, Doug Jones, and anonymous reviewers helped improve this paper. Thanks to Chhay Kong for help with circuit modeling.