Error Mitigation in Digital Logic using a Feedback Equalization with Schmitt Trigger (FEST) Circuit

Zafar Takhirov, Bobak Nazer, and Ajay Joshi
Electrical and Computer Engineering Department, Boston University, Boston, MA, USA
{*zafar, bobak, joshi*}*@bu.edu*

ABSTRACT

Voltage scaling is commonly used to reduce the energy consumption of digital CMOS logic. However, as the supply voltage decreases, transistor switching times increase, leading to intersymbol interference (ISI) between successive outputs of the digital logic. This limits the amount of voltage scaling that can be applied for a target performance. We describe a novel circuit-level technique that couples feedback equalization with a Schmitt trigger (FEST) to suppress this ISI, which in turn enables further voltage scaling while ensuring reliable operation at the desired target performance. For a 4-bit, 22 nm Kogge-Stone adder designed for 2 GHz operation, the proposed technique lowers the critical voltage (beyond which frequent timing errors occur) from 580 mV (nominal design) to 510 mV (design with FEST circuit), providing a 20% decrease in energy per operation. We also apply this technique to a 3-tap 4-bit finite impulse response (FIR) filter operating at 500 MHz, and observe that the critical voltage drops from 680 mV (nominal design) to 510 mV (design with FEST circuit) and the energy per operation can be decreased by up to 40%.

I. INTRODUCTION

Energy consumption has become one of key limiters of the performance of digital complementary metal oxide semiconductor (CMOS) logic systems. Voltage scaling is commonly used for low power operation [1]–[5]. However, decreasing the supply voltage increases the switching times of logic gates. Moreover, voltage scaling increases the hold and setup time of flip-flops and latches. Assuming the target performance is held fixed, these effects make the entire system highly susceptible to timing errors, which are further exacerbated by environmental and process-related variations. Left unchecked, these timing errors can lead to catastrophic application failures. The conventional approach to this problem is to choose the supply voltage conservatively so that the timing constraints of the critical path are not violated.

One possible workaround would be to increase the timing margins by decreasing the number of gates in the critical path to accommodate slower transitions. Unfortunately, this increases power dissipation as more flip-flop stages are needed to implement the same chain of operations. A well-investigated alternative is to allow for the possibility of errors and create mechanisms to correct them when they occur. For instance, error-correcting codes can be used for detecting and correcting multi-bit errors. Similarly, spatial and temporal redundancy, combined with majority voting, can be used to eliminate errors. However, both these approaches are resource-intensive and may even increase the net power consumption.

In this paper, we propose a new feedback-based technique for mitigating timing errors. Many of the timing errors observed after voltage scaling are due to residual effects from the previous computation. In other words, voltage-scaled logic is subject to intersymbol interference (ISI). We develop a Feedback Equalization with Schmitt Trigger (FEST) circuit that adjusts the switching thresholds in the critical path based on prior outputs. Specifically, if the input to a flip-flop is switching, the FEST circuit provides a faster charging (or discharging) path for the input capacitance of the flip-flop to ensure correct sampling of data. Overall, this makes the system more robust to timing errors and enables more power savings through voltage scaling. It should be noted that the FEST circuit can also be used to provide fast charging/discharging paths for the one or more gates in the critical path preceding the flip-flop to further reduce the power dissipation.

The key contributions of this paper can be summarized as follows:

- Taking a 4-bit Kogge-Stone Adder (KSA) as an example, we provide an overview of the timing errors that result from voltage scaling.
- We propose a FEST circuit that provides fast charging/discharging path for the input capacitances of flip-flop and logic gates in the critical path using feedback from the prior computation.
- As a case study, we apply the FEST circuit to the critical paths of two arithmetic logic circuits, a 4-bit KSA and a 3-tap 4-bit finite impulse response

978-1-4673-1036-9/12/$31.00 ©2012 IEEE

(FIR) filter. We demonstrate that the power can be reduced by 20% and 40%, respectively, while mitigating timing errors and maintaining performance.

The rest of the paper is organized as follows. We first present an overview of the other efforts in place to tackle errors in VLSI systems in Section II. Section III provides an analysis of the error profile of a 4-bit KSA after voltage scaling for 22 nm technology. A detailed description of the operation of the FEST circuit is presented in Section IV. Section V discusses two case studies, where we show how the application of the FEST circuit can reduce power dissipation of a 4-bit KSA and 3-tap FIR filter, while maintaining target bit error rates and performance.

II. Related Work

The common approaches for ensuring reliable operation of a VLSI system are to detect and correct errors immediately when they occur, or to make the circuit, the architecture, or the application inherently resilient to errors. There has been a great deal of work on incorporating error-correcting codes into circuit design (see, for instance, [6]–[8]). A mechanism to tolerate errors due to process-voltage-temperature (PVT) variations and soft errors using a circuit-level error detection technique that flags spurious transitions in the stage-holding latch node followed by recovery at the architectural level is proposed in [9]–[11]. In a similar vein, [12] proposes a circuit-level technique that uses a redundant path for error detection and rollback buffers for error correction. Two timing error detection circuits, a dynamic transition detector with a time-borrowing datapath latch (TDTB) and double-sampling static design with a time-borrowing datapath latch (DSTB), that manage the metastability problem (due to voltage and temperature variations) by directing it to the error path are proposed in [13], along with an error recovery circuit that replays the failed instructions at lower frequency. A power-aware slack redistribution that shifts the timing slack of frequently-exercised, near-critical timing paths to enable graceful degradation of design under voltage overscaling is proposed in [14]. Design techniques – dynamic segmentation with multi-cycle error compensation, and delay budgeting for chained data path components, that improve the accuracy of inherently resilient applications under voltage overscaling are proposed in [15].

N-modular redundancy with a voting mechanism has been widely used to tolerate errors. A fluid NMR framework that is aware of the application's inherent error tolerance and uses this knowledge to tradeoff power against reliability is proposed in [16]. In very recent work, [17] has developed a confidence-driven model in conjunction with temporal and spatial redundancy for error detection as well as latency and threshold improvement of digital logic. The error resilient system architecture uses asymmetric reliability of multicore architectures, software optimization and low-overhead checking mechanisms to enhance the performance of probabilistic applications [18]. Similarly, [19] develops a stochastic processing platform that uses multiple functional units that differ in their architecture but share functionality to provide "stochastically correct" execution of error-tolerant applications. At the algorithm level, algorithmic noise tolerance (ANT) schemes that can tolerate errors under voltage overscaling and PVT variations have been proposed in a series of papers [20]–[22].

We propose a circuit-level scheme that uses a Feedback Equalization with Schmitt Trigger (FEST) circuit in the critical path to tolerate errors resulting from voltage overscaling in digital logic. The key feature of this circuit is that it dynamically adjusts the charging/discharging of parasitic capacitors of the gates in the critical path based on the prior outputs. The proposed technique can further lower the power dissipation of the digital logic block while ensuring reliable operation. The proposed technique is independent of the overlying architecture and algorithm, and can be readily incorporated into the digital flow in the design of both low-power custom ASICs and general-purpose processors.

III. Error Manifestations in Digital Logic

A CMOS circuit can have permanent faults (due to irreversible physical changes such as a gate output stuck at one or zero, short, or electromigration-induced open circuits), transient faults (due to temporary changes in supply voltage and temperature) and/or intermittent faults (due to unstable or marginal hardware) [23]. These faults can manifest into bit errors and at times result in catastrophic system failures. In this paper, we will focus exclusively on timing errors due to supply voltage scaling. These timing errors can be traced to the fact that voltage scaling increases the switching time of the transistors, which in turn leads to timing constraint violations that result in incorrect sampling of the data by the flip-flop/latch. We propose circuit-level techniques that can mitigate these errors, thus permitting further voltage scaling and a reduction in energy consumption. For the purposes of this paper, we assume that the application layer is able to tolerate some errors, so long as the probability is very low. Understanding the interaction between our circuit technique and higher

layer error-resiliency is an interesting subject for future study.

For a complex digital logic block with multiple paths between the inputs and outputs, timing violations occur primarily along the critical path. As mentioned earlier, we expect errors to arise when the transitioning outputs of a logic gate do not have sufficient time to charge/discharge due to a drop in supply voltage. To verify this simple hypothesis, we simulated a 4-bit Kogge-Stone Adder (KSA) in HSPICE. We designed the adder to run (without errors) at 2 GHz at a nominal supply voltage of 800 mV at 22 nm technology. We then scaled the supply voltage to 490 mV and simulated the 4-bit KSA using 1000 randomly generated input patterns that were fed in sequentially. We recorded the occurrences of the erroneous outputs. We then took the same 1000 input patterns, reversed their order, fed them sequentially into the adder again and recorded occurrences of the erroneous outputs. As expected, the error occurrences (after reordering) differed for these two experiments, meaning that the errors in a digital block depend on both the current and the prior inputs. An additional observation from this experiment was that the higher order output bits were found to be more error-prone, due to their longer path lengths. Figure 6 shows a plot of the error distribution across the 5 sum bits for a simulation with 1000 randomly generated input patterns.

These observed error manifestations closely mirror the *intersymbol interference* (ISI) errors that arise in communication channels. In this setting, a decision feedback equalization (DFE) mechanism is commonly used to combat ISI at the receiver end. The basic idea is that the receiver subtracts a weighted estimate of the prior bit before making a decision on the current bit. In the next section, we explore a circuit that mimics this technique for use in arithmetic logic.

IV. Equalization techniques

In a communications system, DFE is often implemented by multiplying the estimate of the previous bit by an appropriate scaling factor and subtracting this quantity from the channel output. In our considerations, the multiply-accumulate circuits required for DFE far exceed the complexity of the digital arithmetic logic we are trying to protect. From a digital circuit design perspective, DFE can be viewed as a mechanism to vary the switching threshold of a logic gate based on the prior gate output. If the prior output is a zero, then the switching threshold is decreased to facilitate a transition to a one. Conversely, if the prior output is a one, then the switching threshold is increased. Below, we describe a variable threshold inverter that can be used as a basic feedback equalizer. Afterwards, we argue that coupling this circuit with a Schmitt trigger leads to a robust equalizer that is well-suited for overscaling arithmetic logic to save energy.

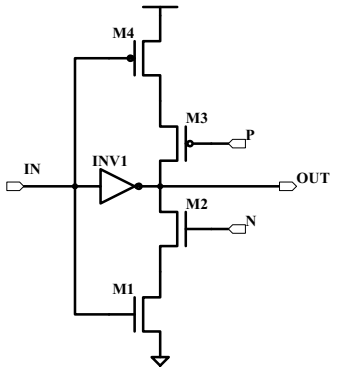

Figure 1: Variable threshold inverter circuit [24].

P	N	V_{th}
LOW	$HIGH$	V_{th}^{0}
LOW	LOW	V_{th}^{+}
$HIGH$	$HIGH$	V_{th}^{-}

Table I: The inverter threshold depends on the inputs P and N. Our system varies the threshold using feedback.

A. Feedback circuit

Sridhara *et al.* employed feedback equalization to overcome the ISI encountered in on-chip communication channels [24]. They used a variable threshold inverter (depicted in Figure 1) to implement feedback equalization on a bus. The switching threshold of this inverter depends upon the values of the inputs at nodes P and N as detailed in Table I with $V_{th}^{-} < V_{th}^{0} < V_{th}^{+}$. The sizes of transistors M1, M2, M3, and M4 are chosen depending on the desired values for V_{th}^{+} and V_{th}^{-}. This circuit can be used as a feedback equalizer by simply connecting the P and N nodes to the previous output sampled by the flip-flop (see Figure 4). In this configuration, the circuit adjusts the switching threshold and facilitates faster high-to-low and low-to-high transitions. Note that the inverter (INV1) should be fairly weak to ensure that the output does not float [24]. The DC response of the variable threshold inverter is shown in Figure 2.

One drawback of using this circuit in low power digital logic is that, in the deep submicron regime, it becomes susceptible to glitches after voltage scaling. This is mainly due to the difference in switching times of the different nodes in the circuit. At nominal voltage,

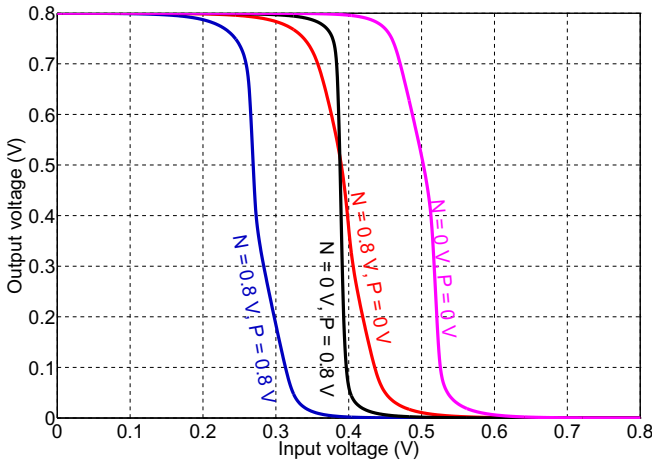

Figure 2: DC response of the feedback equalizer circuit.

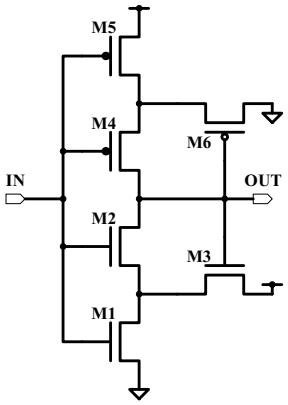

Figure 3: Inverting Schmitt trigger circuit [25].

these glitches can be tolerated and the correct output will be sampled by the succeeding flip-flop. However, these glitches widen with voltage scaling and eventually get incorrectly interpreted as valid signals.

B. Schmitt trigger

The deficiencies of the feedback circuit can be overcome by using a Schmitt trigger. A Schmitt trigger is a dual-threshold buffer or an inverter with positive feedback (i.e. with loop gain greater than one) that can mitigate the impact of glitches. An inverting Schmitt trigger described in [25] is shown in Figure 3. The Schmitt trigger transfer function exhibits a hysteresis loop with two switching thresholds – lower switching point V_{SPL} and higher switching point V_{SPH}. (In a conventional inverter, $V_{SPL} = V_{SPH}$.)

The threshold voltages of the Schmitt trigger can be controlled using the following equations

$$\frac{\beta_1}{\beta_3} = \left[\frac{V_{DD} - V_{SPH}}{V_{SPH} - V_{THN}}\right]^2 \quad (1)$$

where V_{THN} is the threshold voltage of the NMOS and the device transconductances satisfy $\beta_2 \geq (\beta_1$ or $\beta_3)$ (M2 is a switch), and

$$\frac{\beta_5}{\beta_6} = \left[\frac{V_{SPL}}{V_{DD} - V_{SPL} - V_{THP}}\right]^2 \quad (2)$$

where V_{THP} is the threshold voltage of the PMOS and the device transconductances satisfy $\beta_4 \geq (\beta_5$ or $\beta_6)$ (M4 is a switch) [25].

Although the Schmitt trigger is an effective way to eliminate glitches, it exhibits a large delay between the time when the input signal changes and the output signal changes. Thus, despite the fact that the Schmitt trigger has a very short switching time, it can still introduce errors due to a possible shift of the gate output (which would be the input to the flip-flop) transition closer to the clock edge. In non-pipelined, multi-path digital logic blocks, this can cause problems, as flip-flops in the slower computation paths (e.g. the MSB of an adder) may go into a metastable state or might sample the data one clock cycle later than faster paths (e.g. the LSBs).

C. Feedback Equalization with Schmitt Trigger (FEST)

The feedback equalizer described in Section IV-A and the Schmitt trigger in Section IV-B exhibit different output switching behavior for the same input transition. The feedback circuit, due to the altered threshold voltages, starts switching earlier, but the switching time is large due to lower drive strengths (meaning that more time is needed to charge up the parasitics). On the other hand, the Schmitt trigger has a smaller switching time (when the switching starts, $|V_{GS}|$ is already high, and most of the parasitics are charged/discharged), but it starts switching much later. It follows that in the feedback circuit there is a high chance of falling into metastability due to a timing violation. In the case of the Schmitt trigger, the chance of falling into metastability is relatively low (due to fast transitions), but there is a high chance of being sampled at the wrong cycle.

For our proposed FEST circuit we connect the output of the feedback circuit to the input of the Schmitt trigger. The feedback circuit stage reduces the ISI caused by voltage overscaling while the Schmitt trigger stage smooths out any glitches created by the feedback circuit. On their own, the feedback circuit and the Schmitt trigger invert their inputs so the FEST circuit is non-inverting. Overall, the FEST circuit can be viewed as a variable threshold buffer that is robust to ISI and glitches. Figure 4 illustrates how the FEST circuit is connected to enhance pipelined combinational logic. A comparison of the switching thresholds of the four different designs – nominal circuit design, design with

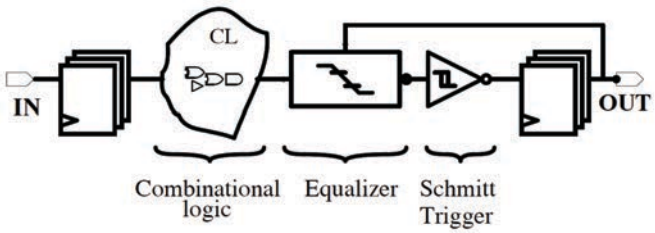

Figure 4: Pipelined combinational logic with FEST circuit..

Input Transition	Nominal design	Feedback Equalizer only	Schmitt Trigger only	FEST
$0 \rightarrow 1$	400 mV	280 mV	600 mV	300 mV
$0 \leftarrow 1$	400 mV	500 mV	200 mV	480 mV

Table II: Comparison of switching threshold voltages for various designs.

feedback equalizer circuit, design with Schmitt trigger circuit and design with FEST circuit, is shown in Table 2. Though the switching threshold of the design with the FEST circuit is higher (lower) than that of the design with feedback equalizer circuit for 0→1 (1→0) input transition, the design with FEST circuit takes advantage of the fast switching of the Schmitt trigger and reaches V_{DD} earlier than the remaining three designs.

V. Experimental Results

We examine two arithmetic circuits – a 4-bit Kogge-Stone adder (KSA) and a 4-bit 3-tap finite impulse response (FIR) filter to determine the effectiveness of our proposed FEST circuit. We compare three designs – nominal arithmetic circuit design, arithmetic circuit design with only the feedback equalizer circuit, and arithmetic circuit design with the FEST circuit. We do not consider the arithmetic circuit design with Schmitt trigger circuit as it does not exhibit graceful degradation in the word error rate (WER) with voltage scaling. For both the KSA and the FIR filter, we add the feedback circuit and the FEST circuit along the critical path(s). We simulate all three designs for the two circuits using the 22 nm predictive technology model (PTM) [26] with a nominal voltage of 800 mV. We scale the supply voltage to determine the tradeoff between WER and energy consumption for all three designs. For each voltage value, we simulate 1000 random operations. In the energy consumption results, clock energy is not included, as the proposed design is independent of the clock energy. To determine computation errors

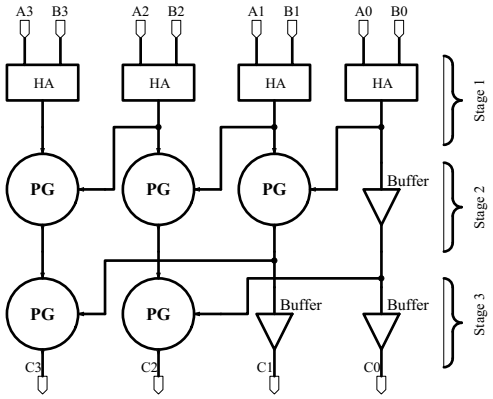

Figure 5: 4-bit Kogge-Stone Adder (KSA) – HA = Half adder, PG = Propagate-Generate unit.

in the KSA and FIR filter, we compare the simulated output with the output of error-free arithmetic logic. The WER is then calculated by dividing the total number of erroneous outputs with the total number of operations.

A. Kogge-Stone Adder

is a parallelized version of a carry look-ahead adder (see Figure 5). It computes its carry signals by calculating propagate (P) and generate (G) signals concurrently in each vertical stage. The result of the adder is generated in the last stage by XORing the output of the PG-calculator. The nominal design of our 4-bit KSA operates at 2 GHz. In Figure 6, we have plotted a histogram of error instances for each of our three designs at 490 mV over 1000 trials. Notice that most of the errors for the nominal and feedback-only designs are concentrated in the higher order bits, which lie on the critical paths. Conversely, the FEST circuit errors are roughly uniform.

Figure 7 shows a 10 ns transient snippet of the most significant bit of the Kogge-Stone Adder, showing the output of the combinational logic (graph A), output of the equalizer (graph B), output of the Schmitt trigger (graph C), and the latched result (graph D). Observe that the output of the KSA combinational logic contains glitches which are the result of different data propagation times. If not filtered out, these glitches cause setup and hold time violations in the output registers. At the output of the feedback equalizer (B), the rise-fall time improves, however some of the glitches are amplified. The Schmitt trigger filters the glitches (C) but slightly delays the signal. However, due to a very low rise-fall time of the Schmitt trigger, it completes its transition earlier then the original signal A, enabling the register to record the results correctly (D). At 520 mV, the average delay and rise/fall times of the FEST circuit are 40ps

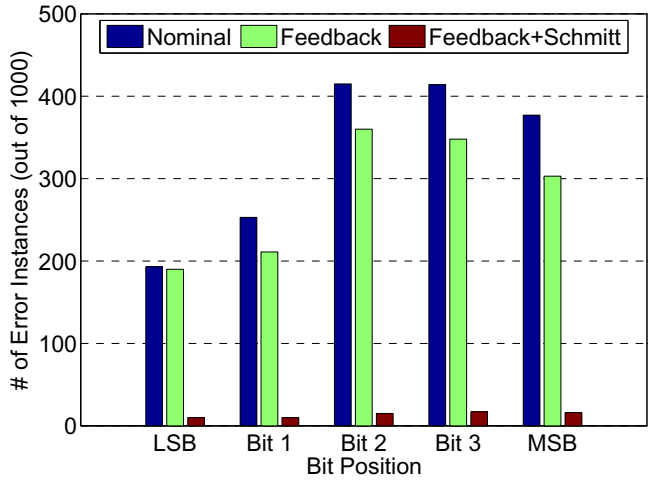

Figure 6: Bit error histogram for 4-bit Kogge Stone Adder designed using 22 nm CMOS technology, running at 2 GHz and operating at 490 mV, and simulated for 1000 randomly generated input patterns.

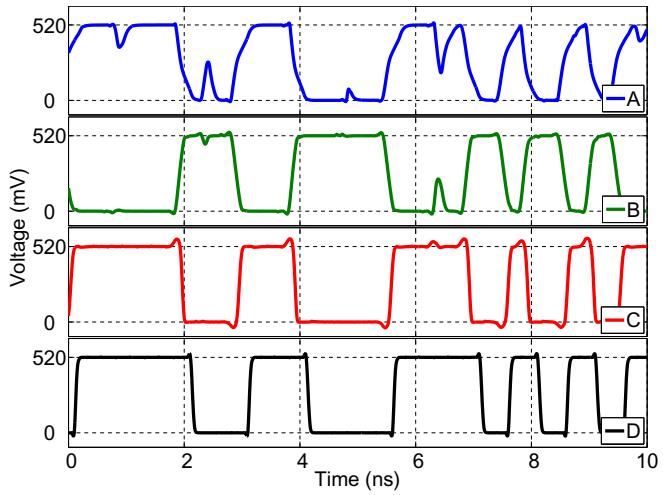

Figure 7: Transient response of the outputs of KSA (A), feedback equalization (B), Schmitt trigger (C), and latched signal (D).

and 70ps, respectively. The KSA without FEST has a rise/fall time of 190ps. If we assume that the rise and fall is linear, the KSA with FEST reaches the 90% point 20ps earlier than the KSA without FEST.

Figure 8 shows the WER of the KSA for the three designs. For the nominal design, errors start to appear below 580 mV. At this point, the charge and discharge times of the transistor parasitic capacitors are large enough to cause interference between computations. The feedback circuit provides comparatively faster charging and discharging paths so it is possible to scale the voltage slightly lower to 570 mV before we start seeing errors due to both glitches and timing

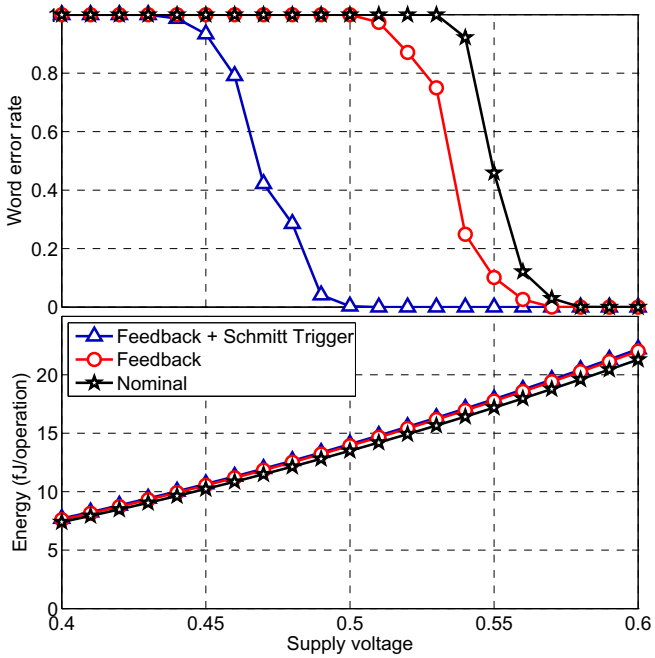

Figure 8: Word error rate and energy consumption for the nominal design, design with only the feedback circuit, and design with the FEST circuit for a 4-bit Kogge-Stone Adder.

errors. The FEST circuit is able to suppress glitches and timing errors until the voltage falls below 510 mV, which leads to significant energy savings.

Figure 8 also shows the energy consumption (data-dependent and fixed) per operation for each design. Note that the energy overhead of the feedback equalizer and FEST circuits is fairly small. To keep the WER near zero, nominal design and design with the feedback equalizer circuit require $\approx$ 19.5 fJ/op. For the same performance, design with the FEST circuit consumes $\approx$ 15.5 fJ/op providing a 20% savings in energy. If an application is error-resilient at the algorithm level and can tolerate a relatively large fraction of errors, the proposed FEST circuit can offer even larger energy savings. For example, at a WER of 0.1, the FEST circuit uses 25% less energy than the other designs.

The area overhead for the KSA example is 27.7%. Note that since the FEST circuit is applied only at the outputs, the area overhead will decrease for larger circuits.

B. Finite Impulse Response Filter

To determine the scalability of the proposed FEST circuit, we consider a 4-bit 3-tap FIR filter. Finite impulse response (FIR) filters are widely used in data communications and audio processing. The output of the FIR filter is defined as the convolution of the input

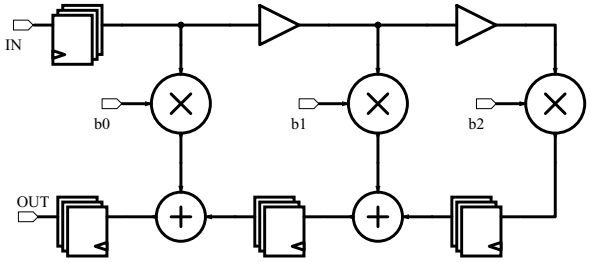

Figure 9: 4-bit 3-tap finite impulse response (FIR) filter.

signal x with an impulse response h. Specifically, the output is given by

$$y[n] = \sum_{i=0}^{N} h_i x[n-i], \tag{3}$$

where $x[n]$ is the input signal, $y[n]$ is the output, h_i is the weight of the i-th tap, and N is the total number of taps (or filter order).

For our case study, we designed and simulated an FIR filter using Wallace multipliers and Kogge-Stone adders as building blocks (see Figure 9 for a block diagram). The critical path, as seen in the block diagram passes through a buffer, multiplier, and an adder. At the nominal supply voltage of 800 mV, the 3-tap FIR filter reliably operates at 500 MHz.

One of the main sources of errors for this filter implementation is that the input signals to the adders arrive at different times – out of the two input signals, one of the signals passes through a multiplier before arriving at the input of the adder, while the second signal has no such delay. This causes the output signal of the adder to have a glitch. At nominal voltage, this glitch has a small width and does not get sampled by the flip-flop following the adder. However, as the supply voltage is scaled down, the width of the glitch increases, and at a critical voltage it is incorrectly sampled and interpreted as an independent bit value by the flip-flop following the adder. The FEST circuit enables us to further scale down the supply voltage while maintaining reliable operation by mitigating the errors resulting from glitches and slow transitions.

In Figure 10, we have plotted the WER for each of the three designs operating at 500 MHz. For the nominal design, design with only the feedback equalizer circuit, and design with FEST circuit, the errors start to appear below 680 mV, 610 mV, and 510 mV, respectively. Figure 10 also shows the energy consumption of the 3-tap FIR filter for the three designs. To keep the WER near zero, the energy consumed by the nominal design, design with only the feedback equalizer circuit,

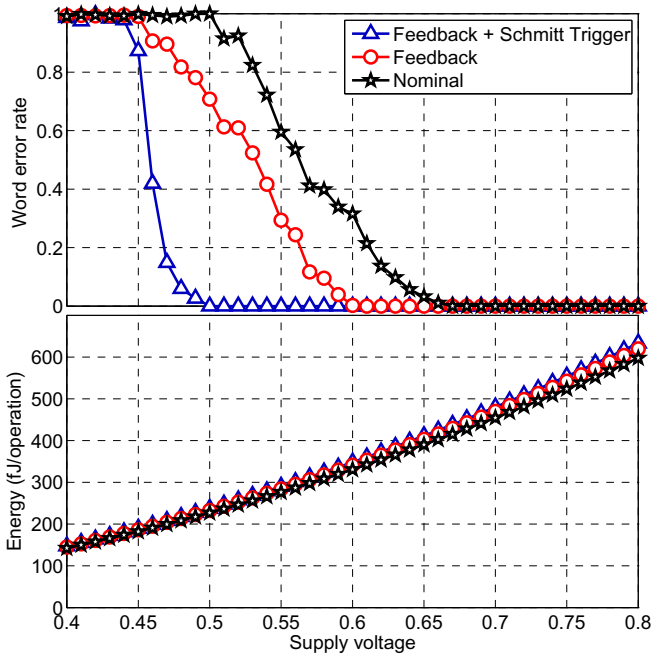

Figure 10: Word error rate and energy consumption for the nominal design, design with only the feedback circuit, and design with the FEST circuit for a 3-tap FIR filter.

and design with FEST circuit is 420 fJ/op, 350 fJ/op, and 250 fJ/op, respectively. Thus, the design with only the feedback equalizer circuit and design with the FEST circuit can provide 16% and 40% energy savings. If the overlying algorithm is error resilient up to a 10% error rate, then we get 5.7% energy savings with only the feedback circuit and 34.3% energy savings. The area overhead for the FEST circuit is 15.8%.

VI. Conclusion

We proposed a feedback equalization with Schmitt trigger (FEST) circuit that can be used to mitigate timing errors in digital CMOS logic. In particular, we focused on timing errors due to elongated transistor switching times that arise under aggressive voltage scaling. The FEST circuit simultaneously reduces switching thresholds (through feedback equalization) and decreases switching times (via a Schmitt trigger). This enables a rapid switching of the logic gate outputs when inputs change and provides the opportunity for further scaling of supply voltage to reduce energy in digital logic designs. As a case study, we implemented a 4-bit Kogge Stone adder (KSA) and a 3-tap 4-bit finite impulse response (FIR) filter in 22 nm CMOS technology. For the KSA, we obtained 20% reduction in energy per operation while maintaining an operating frequency of 2 GHz with no errors over 1000 trials.

For the FIR filter, we obtained up to a 40% reduction in energy, while maintaining the operating frequency at 500 MHz with no errors over 1000 trials.

REFERENCES

[1] J. Howard and et. al, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 173 –183, Jan. 2011.

[2] Y.-H. Lee and et al, "A DVS embedded power management for high efficiency integrated SoC in UWB system," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 11, pp. 2227 –2238, Nov. 2010.

[3] B.-G. Nam and H.-J. Yoo, "An embedded stream processor core based on logarithmic arithmetic for a low-power 3-d graphics soc," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 5, pp. 1554 –1570, May 2009.

[4] M. Sinangil, N. Verma, and A. Chandrakasan, "A reconfigurable 8T ultra-dynamic voltage scalable (U-DVS) SRAM in 65 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 11, pp. 3163 –3173, Nov. 2009.

[5] V. Sze, D. Finchelstein, M. Sinangil, and A. Chandrakasan, "A 0.7-V 1.8-mW H.264/AVC 720p video decoder," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 11, pp. 2943 –2956, Nov. 2009.

[6] P. K. Lala, *Self-Checking and Fault-Tolerant Digital Design*. San Diego, CA: Academic Press, 2001.

[7] J. Mathew, J. Singh, A. Taleb, and D. Pradhan, "Fault tolerant reversible finite field arithmetic circuits," in *On-Line Testing Symposium, 2008. IOLTS '08. 14th IEEE International*, Jul. 2008, pp. 188 –189.

[8] M. Poolakkaparambil, J. Mathew, A. Jabir, D. Pradhan, and S. Mohanty, "BCH code based multiple bit error correction in finite field multiplier circuits," in *Quality Electronic Design (ISQED), 2011 12th International Symposium on*, Mar. 2011, pp. 1–6.

[9] D. Ernst and et al, "Razor: circuit-level correction of timing errors for low-power operation," *Micro, IEEE*, vol. 24, no. 6, pp. 10 –20, Nov.-Dec. 2004.

[10] S. Das and et al, "A self-tuning DVS processor using delay-error detection and correction," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 4, pp. 792 – 804, Apr. 2006.

[11] ——, "RazorII: In situ error detection and correction for PVT and SER tolerance," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 32 –48, Jan. 2009.

[12] H. Naeimi and A. DeHon, "Fault-tolerant sub-lithographic design with rollback recovery," *Nanotechnology*, vol. 19, no. 11, p. 115708, 2008. [Online]. Available: http://stacks.iop.org/0957-4484/19/i=11/a=115708

[13] K. Bowman and et al, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 49 –63, Jan. 2009.

[14] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in *Design Automation Conference (ASP-DAC), 2010 15th Asia and South Pacific*, Jan. 2010, pp. 825 –831.

[15] D. Mohapatra, V. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, Mar. 2011, pp. 1 –6.

[16] J. Sartori, J. Sloan, and R. Kumar, "Fluid NMR – performing power/reliability tradeoffs for applications with error tolerance," in *SOSP Workshop on Power Aware Computing and Systems*, 2009.

[17] C.-H. Chen and et al, "A confidence-driven model for error-resilient computing," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, Mar. 2011, pp. 1 –6.

[18] L. Leem and et al, "ERSA: Error resilient system architecture for probabilistic applications," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, Mar. 2010, pp. 1560 –1565.

[19] S. Narayanan, J. Sartori, R. Kumar, and D. Jones, "Scalable stochastic processors," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, Mar. 2010, pp. 335 –338.

[20] R. Hegde and N. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *Low Power Electronics and Design, 1999. Proceedings. 1999 International Symposium on*, 1999, pp. 30 – 35.

[21] R. Abdallah and N. Shanbhag, "Minimum-energy operation via error resiliency," *Embedded Systems Letters, IEEE*, vol. 2, no. 4, pp. 115 –118, Dec. 2010.

[22] ——, "Error-resilient low-power viterbi decoder architectures," *Signal Processing, IEEE Transactions on*, vol. 57, no. 12, pp. 4906 –4917, Dec. 2009.

[23] C. Constantinescu, "Trends and challenges in VLSI circuit reliability," *IEEE Micro*, vol. 23, no. 4, pp. 14–19, Jul.-Aug. 2003.

[24] S. Sridhara, G. Balamurugan, and N. Shanbhag, "Joint equalization and coding for on-chip bus communication," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, no. 3, pp. 314 –318, Mar. 2008.

[25] R. J. Baker, *CMOS Circuit Design, Layout, and Simulation, Second Edition*. Wiley-IEEE Press, 2004.

[26] (2011). [Online]. Available: http://ptm.asu.edu/