

# A Preliminary Look at Error Avoidance in Digital Logic via Feedback Equalization

Zafar Takhirov, Bobak Nazer, and Ajay Joshi

**Abstract**—This note describes preliminary efforts to incorporate feedback into error avoidance schemes for combinatorial efforts. Taking a 4-bit Kogge-Stone adder as a case study, we develop a feedback equalization scheme that cuts down on errors without increasing energy consumption.

## I. INTRODUCTION

The classical approach to digital circuit design relies on a type of separation principle: the devices are built to be ultra-reliable and the architecture is designed with error-free logic in mind. Unfortunately, the required energy and production costs for this design approach are becoming increasingly unsustainable as feature sizes continue to scale down. One possible solution is allow individual devices to operate in a “noisy” regime and develop a global architecture that is robust to the resulting errors. At a first glance, error-correcting codes seem like a natural fit for this problem. Unfortunately, the power and area overhead required for an error-correcting code often dwarfs the savings.

In a communication system, feedback often requires a significant expenditure of resources (e.g. power, bandwidth) whereas in a circuit, feedback is simply a matter of adding another wire. Given that feedback can reduce the complexity of a coding scheme for reliable communication, it is reasonable to assume that it can also reduce the complexity for reliable computation in a circuit. This note is a preliminary account of our effort to develop feedback-based error management strategies for combinatorial logic. As a case study, we consider a 4-bit adder and develop a simple decision feedback equalization scheme.

## II. RELATED WORK

There has been a great deal of work on error-correcting codes for digital logic over the past few decades. A full accounting of the literature is beyond the scope of this note (see, for instance, [1]). A sequence of papers by Shanbhag and co-authors has set forth a framework for detecting and mitigating errors in digital logic [2]–[4]. The RAZOR architecture clocks digital logic faster than its critical path by catching errors using shadow latches and restarting the downstream pipeline [5]. The Error Resilient

This work was supported by a 2011 Dean’s Catalyst Award from the College of Engineering at Boston University.

The authors are with the Department of Electrical and Computer Engineering, Boston University, Boston, MA. Emails: {zafar,bobak,joshi}@bu.edu

System Architecture (ERSA) uses a combination of reliable and error-prone processors to reduce energy consumption while maintaining reliability [6].

## III. CASE STUDY: 4-BIT ADDER

A 4-bit adder computes the sum of its input bits  $\mathbf{a} = [a_0 \ a_1 \ a_2 \ a_3]$  and  $\mathbf{b} = [b_0 \ b_1 \ b_2 \ b_3]$  plus a carry bit  $c_{IN}$ . The output is given by  $\mathbf{s} = [s_0 \ s_1 \ s_2 \ s_3]$  plus a carry bit  $c_{OUT}$ . Recall that each output bit can be written as

$$s_i = a_i \oplus b_i \oplus c_{i-1}$$

where  $c_{i-1}$  is the carry from the previous sum,

$$c_i = (a_i \cdot b_i) \oplus (c_{i-1} \cdot (a_i \oplus b_i)).$$

Define  $c_{OUT} = c_4$  and  $c_{IN} = c_0$ .

We simulated a 4-bit Kogge-Stone adder [7] in HSPICE with a feature size of  $22nm$ . Transistors were minimum-sized such that the critical path could meet the timing constraint of a  $2GHz$  clock at a nominal supply voltage of  $800mV$ . We studied the effects of lowering the supply voltage below  $800mV$  on the error rate and designed a feedback equalization circuit to reduce the resulting errors.

### A. Error Model

Our only source of error stems from the reduced supply voltage and its impact on the simulated circuit. That is, we did not account for external sources of transient noise in this initial study. To determine how the errors depend on the inputs, we generated 1000 random inputs sequences  $(\mathbf{a}, \mathbf{b}, c_{IN})$  and collected the resulting outputs at a supply voltage of  $510mV$  (while maintaining the clock speed). We observed that approximately 4.5% of the computations were in error. Moreover, we noticed that the errors were not memoryless. Specifically, the same 1000 input patterns, when run in reverse, resulted in a distinct error pattern.

This error memory can be attributed to the fact that the transistors do not have enough time to charge and discharge when the supply voltage is decreased. Therefore, the previous computation may be a source of error to the current one.

### B. Decision Feedback Equalization

A similar source of errors arises in communications settings in the form of intersymbol interference (ISI) in channels with memory. Here, the solution is to remove the

residual effects of the previous bit from the observed signal prior to making an estimate of the current bit. This can be done using the decoder's estimate of the previous bit. Our goal is to mimic this behavior in our adder. That is, if the previous output bit is 0, then the feedback circuit should make it easier to transition to 1 for the upcoming output bit. Similarly, if the previous bit is a 1, then it should be made slightly easier to transition to 0.

In Figure 1, we have drawn our feedback equalization circuit acting on a single inverter. The first two transistors on the left are an inverter for which the four remaining transistors act as a feedback equalizer. This is the same circuit that was used for equalization in on-chip communication by [8]. To the best of our knowledge, this paper is the first to apply this technique to an arithmetic block.

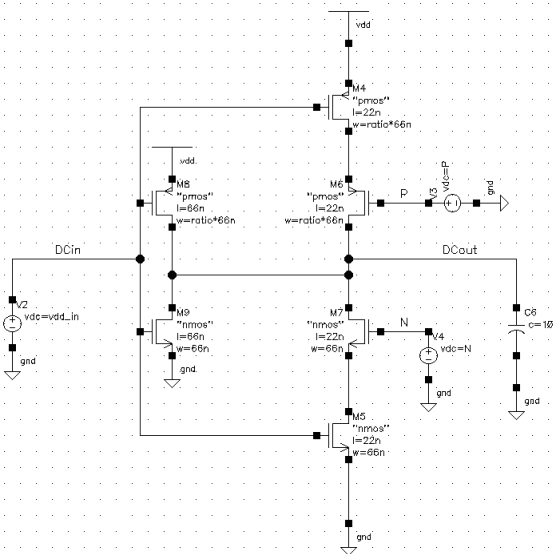


Fig. 1. Feedback equalization circuit applied to an inverter.

If the inputs  $P$  and  $N$  are set to 0 and 1, respectively, then the circuit behaves like a regular inverter. If  $P = N = 0$ , then the threshold is slightly increased and if  $P = N = 1$ , then the threshold is slightly decreased. If we hook up the output of a register that retains the previous value to  $P$  and  $N$ , then this circuit has exactly the desired effect. Specifically, if the previous value is 0, then the threshold will be slightly higher. Therefore, if the input transitions from a 1 to a 0, the threshold is closer to 1 and will be crossed faster than it would be without the feedback circuit. A similar process speeds up transitions between 0 and 1. Of course, this does slightly increase the time needed between identical outputs but since this time is much smaller than the time needed between transitioning outputs.

This feedback equalization circuit was applied to the two most significant bits of our Kogge-Stone adder. We then applied 1000 random input patterns and found that we could lower the supply voltage by an additional  $10mV$  while maintaining the same error rate. At this lower voltage, the feedback-based adder consumes roughly the same amount of

power as standard adder operating at a  $10mV$  higher supply voltage. Thus, our scheme's main benefit is that it adds some additional robustness to supply voltage fluctuations.

#### IV. FUTURE WORK

In this note, we have described a feedback equalization scheme for digital logic and explored its benefits for a 4-bit Kogge-Stone adder. Our future work will focus on developing more advanced feedback techniques that offer higher energy savings.

#### REFERENCES

- [1] P. K. Lala, *Self-Checking and Fault-Tolerant Digital Design*. San Diego, CA: Academic Press, 2001.
- [2] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *ISLPED*, 1999.
- [3] N. R. Shanbhag, "Reliable and efficient system-on-chip design," *Computer*, vol. 27, pp. 42–50, March 2004.
- [4] N. R. Shanbhag, R. A. Abdallah, R. Kumar, and D. L. Jones, "Stochastic computation," in *DAC*, (New York, NY), 2010.
- [5] 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*, (San Diego, CA), Dec. 2003.
- [6] L. Leem, H. Cho, J. Bau, Q. A. Jacobson, and S. Mitra, "ERSA: Error resilient system architecture for probabilistic applications," in *DATe*, (Dresen, Germany), Mar. 2010.
- [7] P. Kogge and H. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Transactions on Computers*, vol. 22, pp. 783–791, August 1973.
- [8] S. R. Sridhara, G. Balamurugan, and N. R. Shanbhag, "Joint equalization and coding for on-chip bus communication," *IEEE Transactions on VLSI Systems*, vol. 16, pp. 314–318, March 2008.