Design of Error-Resilient Logic Gates with Reinforcement Using Implications

Xijing Han, Marco Donato, R.Iris Bahar, Alexander Zaslavsky, William Patterson
School of Engineering
Brown University
Providence, RI 02906

ABSTRACT

Operating circuits in the sub-threshold region can save power, but at the cost of higher susceptibility to noise. This paper analyzes various gate-level error-mitigation designs appropriate for sub-threshold circuits. Previous works have proposed a modified version of the Schmitt trigger gate that uses logic implications to reinforce correct functional behavior. However, the increased error resilience requires increased area, delay, and power overhead. To address these shortcomings, we introduce two alternative and less costly approaches to reinforcing correct logic behavior via implications. In addition, to provide more flexibility in implication selection, we consider not just simple implications that reinforce relationships between two signals, but also more complex 3-signal implications within the circuit. Our simulation results demonstrate that these alternative gate structures can outperform the Schmitt trigger version as long as the noise on the reinforcement signals themselves is sufficiently low.

Keywords

sub-threshold circuit; noise immunity; logic implication

1. INTRODUCTION

Operating circuits in the sub-threshold region minimizes energy consumption, but also causes the circuit to be much more susceptible to noise. Past approaches for error mitigation in combinational circuits used techniques such as triple modular redundancy [4] or duplicating only a portion of the circuit [10], but required significant area and power overhead. Other approaches, such as [17], selectively target nodes that are most susceptible to noise and increase the transistor sizes of the gates driving these nodes in order to achieve higher noise immunity. While the approach of [17] is reported to achieve 90% coverage for soft errors, average overhead is 38%, 27%, and 4% respectively in area, power, and delay. Finally, approaches based on probabilistic methods such as Markov Random Field gates [13] or Turtle Logic [9] have been proposed for mitigating the effects of noise. While there are many potential sources of noise, from current noise due to charge trapping in transistors to crosstalk voltage noise, in this paper we will focus on thermal noise as a fundamental, technology-independent noise source that has a clear statistical description. Thermal noise is expected have a significant impact in determining the fault rate in future low-power [11] and especially sub-threshold circuits, so new solutions that provide error mitigation at lower cost are needed.

Logic implications have been effectively used for online error detection [2], [3], [12], as well as error correction [1], [8]. As was shown in [8], the correction can be implemented by translating the logic relationship described by an implication using a modified version of a Schmitt trigger gate [5]. While the modified Schmitt gate was shown to have some useful error correction properties, it can only mitigate errors caused by noise on the inputs of the gate. That is, if there is a noise spike on the output of the gate, the Schmitt gate cannot mitigate this error. In addition, the extra series transistors in the Schmitt gate can lead to increased switching delays.

The method proposed in [1] adds redundant source wires as inputs to gates to correct erroneous values on a target wire. While this approach has modest area overhead, the usage of implications between source and target wires to mitigate errors is constrained: the value on the source wire may be masked by other input signals on the gate, so even when the implication is active, the error on the target wire may not be corrected.

To extend implication-based error correction beyond prior work, in this paper we investigate two alternative gate structures to reinforce correct behavior in the presence of thermal noise. Our two new gate designs provide more flexible use of implications than either [1] or [8] by considering both single-node and 2-node implications. That is, the implication can consist of a single source signal (also called the *control* signal) or two sources. Our new gates also require less transistor overhead than the modified Schmitt gate [8].

We compare these reinforcement gates in terms of area, power, delay, and noise immunity. Our simulation results show improved noise immunity with 2-node implications. In addition, we show that our gates can provide better error correction than the Schmitt gate when noise appears at the output of the gate. However, the Schmitt gate remains an attractive choice when the controlling signals for the implication are themselves susceptible to significant noise.

The remainder of this paper is organized as follows. Sec-

GLSVLSI '16, May 18 - 20, 2016, Boston, MA, USA

ISBN 978-1-4503-4274-2/16/05...$15.00

DOI: http://dx.doi.org/10.1145/2902961.2902983

tion 2 provides the background on logic implications and how they may be used for error mitigation in combinational circuits. In Section 3, we propose three versions of logic gates with reinforcement implemented to mitigate errors and analyze their advantages and disadvantages. In Section 4, we discuss the optimal selection of implications. Experimental results in terms of delay, power and noise resiliency are provided in Section 5, where we also discuss the impact on performance of sizing the control transistors. Concluding remarks and future directions are given in Section 6.

2. BACKGROUND

The proposed method is based on the existence of logic implications, or invariant relationships, between signal nodes in a circuit. In this paper, we propose two types of implications: 1-node-implications and 2-node-implications.

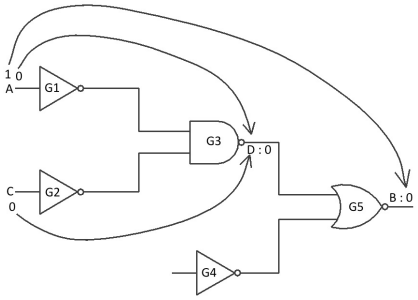

Figure 1: Illustrative circuit for defining 1-node and 2-node implications. Implications shown are $[A:1 \rightarrow B:0]$ and $[A:0, C:0 \rightarrow D:0]$

2.1 1-node-implications

The 1-node-implication between nodes A and B is illustrated in Figure 1 and can be expressed in the form of [A: 1] →[B: 0], where A is the *source*, or *control* signal and B is the *target*. There exist four types of logic value implications: [0→0], [0→1], [1→0] and [1→1]. As will be clarified in Section 4, based on the error-mitigation performance of our reinforcement gates, we prefer to use [0→1] and [1→0]. If we want to reinforce implications [1→1] and [0→0] we have to add an inverter to transform the implication to the desired form; this requires additional area and delay overhead, so we try to avoid these implications as much as possible.

2.2 2-node-implications

The 2-node-implication uses a combination of two nodes as the controlling signals to reinforce the output of a gate. The 2-node-implication is similar to the multi-site implication proposed in [3], but in our case is used for error correction. In Figure 1, the implication between nodes A, C and D is an example of a 2-node-implication. In this case, we can express this implication is in the form of [A: 0], [C: 0] → [D: 0], where A and C are the control signals and D is the target. For the same reason given for 1-node-implications, we prefer using [1,1→0] and [0,0→1] implications for reinforcement. However, we can use [0,0→0] and [1,1→1] implications by adding a NAND gate or NOR gate to transform the 2-node-implication to a 1-node-implication. For example, by adding a NOR gate with inputs A and C from Figure 1, the output of the NOR gate can be used to realize the implication $[NOR:1 \rightarrow D:0]$.

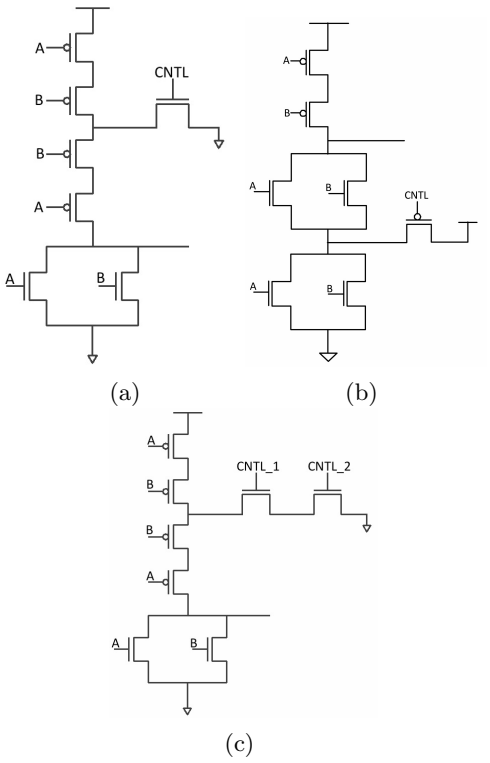

Figure 2: Modified Schmitt gates using: (a) 1-node-implications ($CNTL:1 \rightarrow Out:0$) (a) and ($CNTL:0 \rightarrow Out:1$) (b); (c) 2-node-implication ($CNTL1:1, CNTL2:1 \rightarrow Out:0$).

The 2-node-implication provides more flexibility to choose reinforcement options. For example, when implication sets [0→1] or [1→0] cannot be found, we can use [0,0→1] or [1,1→0] instead. The 2-node-implication also has higher reliability than the 1-node-implication due to error-masking, since noise on either control signal alone is less likely to incorrectly alter the output of the gate.

3. LOGIC GATES WITH REINFORCEMENT

Logic gate reinforcement with implications uses a controlling signal to bias or force the value of the output. In this section, we propose three versions of reinforcement gates: Schmitt trigger, Direct Control (DC) and Complementary Direct Control (CDC).

3.1 Schmitt gate

The version of Schmitt gate considered in this work was originally presented in [8]. Figure 2(a) shows the architecture of a 2-input NOR Schmitt gate with implication $CNTL:1 \rightarrow Out:0$. Figure 2(b) shows the architecture of a 2-input NOR Schmitt gate with implication $CNTL:0 \rightarrow Out:1$. This specific implication type allows us to highlight some important aspects of the noise response of this gate. Applying the reinforcement on the pull-up network of a NOR gate requires having four transistors in series to the output, causing a significant increase in delay. While this solution has a negative impact on the circuit performance, it mitigates the effects of noise at the input. Applying the reinforcement on the pull-down network of a NOR gate, as

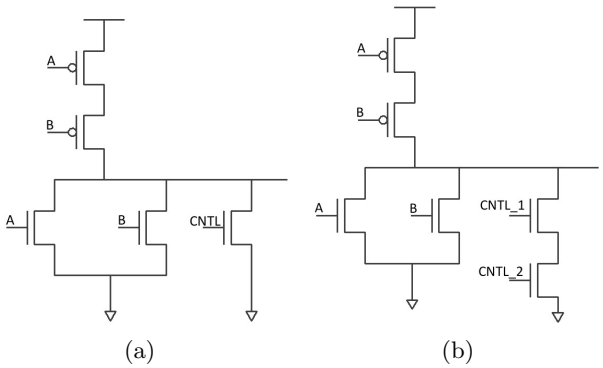

Figure 3: The Direct Control gate using (a) 1-node-implication and (b) 2-node-implication.

shown in Figure 2(b), requires having two transistors in series which has a smaller delay. However, the area overhead for this gate does not scale well with the number of inputs. Another critical aspect of the Schmitt gate circuit is that the control branch is not directly tied to the gate's output. As a consequence, the Schmitt gate can attenuate the input noise but cannot prevent noise at the output from impacting the gate's logic behavior. Figure 2(c) shows the architecture of the Schmitt gate implementing a 2-node-implication, which faces the same drawbacks as the 1-node versions.

3.2 Direct Control (DC) gate

To address the relatively high area overhead of the Schmitt gate and its inability to mask noise at the output, we investigated an alternative method for adding implication reinforcement to logic gates. The DC gate uses transistors to connect the output and the control branch directly. The additional transistors are controlled by the source signals of the implication. Figure 3(a) shows the DC implementation for a 2-input NOR gate with implication $CNTL : 1 \rightarrow Out : 0$. In this case, we are forcing rather than biasing the value at the output based on the implication values. The DC gate has advantages in terms of size and speed; in particular, the transistor overhead is independent of the number of inputs. However, it is sensitive to the noise on the control signal (i.e., the implication source node) since this signal affects the output directly. Figure 3(b) shows the architecture of a Direct Control gate implementing a 2-node-implication.

3.3 Complementary Direct Control (CDC) gate

The DC gate in Section 3.2 uses the implication in either the pull-up or the pull-down network. Therefore, when the implication is actively correcting an error on one polarity (*i.e.,* either V_{DD} or ground), the network connected to the opposite polarity could introduce contention at the output node. To make sure that only one network can determine the value of the output, we propose a CDC gate that was previously used in [1]. Figure 4(a) shows the architecture of the CDC gate. When the implication is active, the control signal will reinforce one polarity while cutting off the connection to the other polarity. The drawback is that this implementation is even more sensitive to noise on the control signal, with the risk of reinforcing the wrong value at the output. Figure 4(b) shows the architecture of the CDC gate implementing a 2-node-implication.

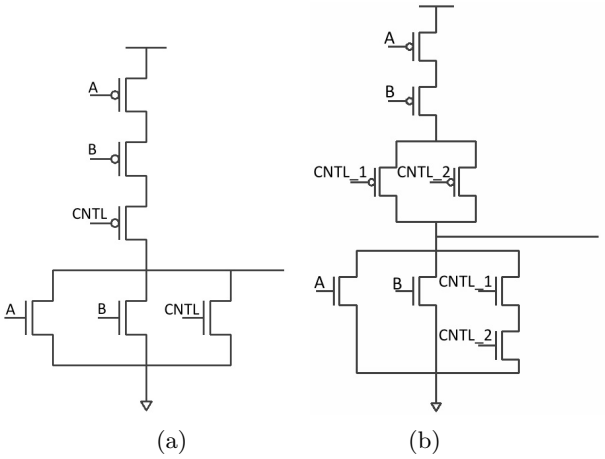

Figure 4: The Complementary Direct Control gate using (a) 1-node-implication and (b) 2-node-implication.

4. IMPLICATION SELECTION

Once a full list of implications in a circuit is determined (e.g., by using the implication discovery procedure proposed in [2]), it is not necessary or feasible to use all of them to reinforce the circuit. Instead, we follow a set of rules to select an optimized set of implications. As mentioned in Section 2, there exist four types of 1-node-implications: [0→0], [1→0], [0→1],[1→1]. Note that for MOSFET transistors, an NMOS passes a weak V_{DD} signal and a PMOS passes a weak zero voltage, so using an NMOS on the pull-up path to reduce the noise in a V_{DD} signal results in a poorer error-mitigation performance, compared to using a PMOS. Therefore, for the four types of implications, [0→1] and [1→0] are our first choice. For the same reason, for 2-node-implications, [1,1→0] and [0,0→1] are preferred.

An implication consists of source (i.e., control) node and target node. The target node can be of 2 different types:

- An "essential node" such as a primary output.
- A node that is easily affected by noise, such as the output of an inverter (i.e., errors on the input cannot be masked)

Once the target node is decided, the next step is choosing its corresponding control node. This control node should be reliable, have a high probability of being active, and be at a distance of more than one gate from the target node. The steps to select the control node are as follows:

1. For all implications involving the target node, calculate the probability of the implication being active (i.e., the probability the control node is at the appropriate logic value) and pick the implications with the highest activation probability.
2. Among the high-activation implications chose the one with the most reliable control node. As a simple initial metric, in this paper we assume that nodes closer to the primary inputs have higher reliability.

1-node-implication				
	implication 1→0		implication 0→1	
	Active	Inactive	Active	Inactive
Schmitt gate	7.43ns	6.78ns	5.29 ns	5.38ns
DC gate	2.98ns	4.57ns	2.69ns	4.69ns
CDC gate	4.09ns	5.05ns	3.62ns	5.04ns
2-node-implication				
	implication 11→0		implication 00→1	
	Active	Inactive	Active	Inactive
Schmitt gate	8.37ns	6.36ns	4.88ns	5.57ns
DC gate	4.60ns	5.10ns	4.55ns	5.40ns
CDC gate	5.53ns	6.19ns	6.54ns	5.48ns

Table 1: Delays of three versions of 2-input NOR gate with reinforcement

1-node-implication				
	implication 1→0		implication 0→1	
	Active	Inactive	Active	Inactive
Schmitt gate	3.01nW	2.51nW	2.60nW	1.74nW
DC gate	2.24nW	2.12nW	2.17nW	1.57nW
CDC gate	2.18nW	2.02nW	2.15nW	1.53nW
2-node-implication				
	implication 11→0		implication 00→1	
	Active	Inactive	Active	Inactive
Schmitt gate	3.08nW	2.36nW	2.74nW	1.48nW
DC gate	2.39nW	1.35nW	2.37nW	1.38nW
CDC gate	2.48nW	2.00nW	2.42nW	1.56nW

Table 2: Power of three versions of 2-input NOR gate with reinforcement

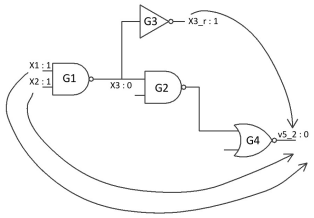

Figure 5: A portion of the *rd53* circuit.

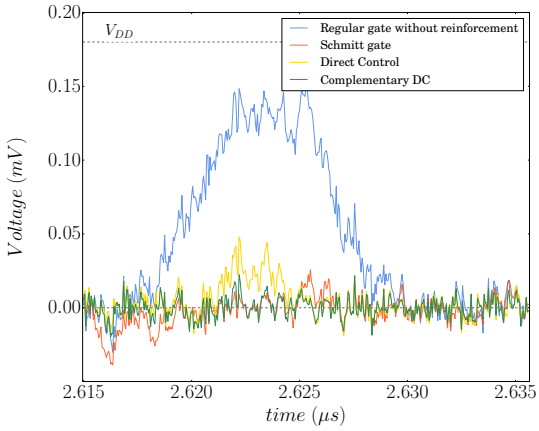

Figure 6: Simulation of v5_2 with active implication using different versions of gates (error caused by noise in the input of gate G4).

5. EXPERIMENTAL RESULTS

5.1 Simulation set-up

The proposed circuits were simulated in SPICE using a 7nm FinFET Predictive Technology Model [15]. The supply voltage was set to $V_{DD} = 180mV$, which is below the threshold voltage of $\sim 500mV$ for both PMOS and NMOS transistors. Thermal noise was modeled using current pulse time series generated using the method presented in [6]. We used this approach as it allows to generate accurate noise samples for sub-threshold circuits at a reasonable computational cost. The RMS noise voltages of minimum-size inverters and two-input gates in the 7nm technology operated at $V_{DD} = 180mV$ are $\sim 5.88mV$ and $\sim 5mV$, respectively [6]. In order to emphasize the characteristic noise response of each of the circuits under analysis, we applied a 12× gain to all noise traces. This leads to highly exaggerated noise, but permits the evaluation of noise resilience and error mitigation within a reasonable computation time in SPICE. More realistic thermal noise transients with unamplified noise were presented in [6] and a custom simulator of rare errors due to unamplified noise in [7], but standard SPICE simulation requires noise amplification. The amplified uncorrelated thermal noise sources were connected to the output node of each gate. All primary inputs were sent through buffers to apply the same noise to those signals as well.

5.2 Delays and power

The delays and power results were calculated for a 2-input NOR gate using three versions of logic gate with reinforcement. A minimum-sized inverter was connected at the output of the NOR gate. Tables 1 and 2 summarize the performance of the three versions of gates in terms of delays and power. The Schmitt gate has larger delay and power dissipation than the other two gates for both the 1-node or the 2-node implications. There is no significant difference between the DC gate and CDC gate in terms of delay and power. In most cases, delays and power dissipation of the 2-node-implication method are both slightly higher than the 1-node-implication method.

5.3 Error-correcting behavior

The experimental results come from *rd53*, a circuit from MCNC benchmark suite [16]. To illustrate the performance of three versions of gates using different kinds of implications, we use implications to reinforce the node v5_2, which is the final output, when the output is low. The relevant portion of the *rd53* circuit is shown in Figure 5.

Figures 6 and 7 show a period of a transient simulation for v5_2 using three versions of gates with the active implication [x3_r:1→v5_2:0]. The ideal output should stay at zero, but the output occasionally drifts to a high voltage because of noise. The error in Figure 6 is caused by the noise in the inputs of G4. If there is no implication reinforcement for v5_2, the maximum value of the error is around 148mV. All of the three versions of reinforcement gates mitigate the error. However, the DC gate has the worst error-mitigation performance when the error is caused by the input, with the nominally low output reaching a maximum value of 48mV. The Schmitt gate has a better error-mitigation performance because it can bias the input-output transfer function towards the correct output value. The CDC gate also has

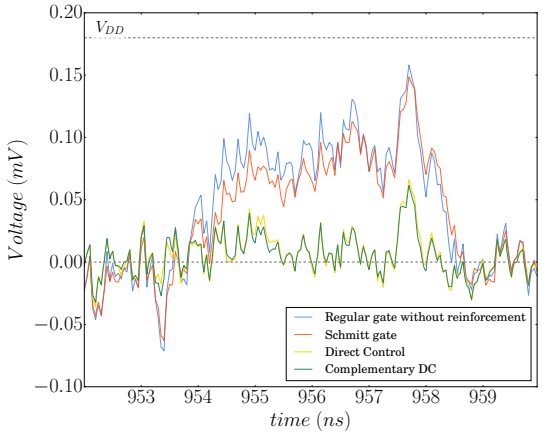

Figure 7: Simulation of v5_2 with active implication using different versions of gates (error caused by noise added at v5_2).

a better error-mitigation performance because it eliminates the contention between the two paths.

The error in Figure 7 is caused by the noise added at v5_2. If there is no implication reinforcement for v5_2, the maximum value of the error is around 158mV. The maximum value of the Schmitt gate, DC gate and CDC gates are 148mV, 66mV and 61mV respectively. Note that the control signal of the Schmitt gate does not directly create a path to the output of the gate, and since the noise is added directly at the output, the Schmitt gate is ineffective at attenuating the error. The other two versions still work to mitigate the error since both of them force the value at the output directly. By adjusting the size of control transistor, the 2-node-implication method can achieve the same error-correcting performance as the 1-node-implication method, which is demonstrated by Figure 8.

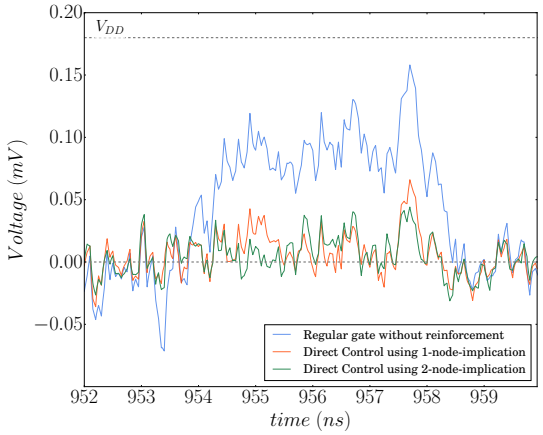

Figure 8: Simulation of v5_2 comparing 1-node and 2-node implications for the DC gate. Implications are active in both cases, with error caused by noise on the output node.

Figure 9 shows a period of transient simulation for v5_2 using three versions of gates with the inactive 1-node-implication [x3_r:1→v5_2:0] and the DC gate with inactive 2-node-implication [x0:1,x1:1→v5_2:0]. In this simulation, the ideal output should stay at 180 mV but the output jumps to low voltage values at times because of noise in the control node. The implication [x3_r:1→v5_2:0] should not be active when the control node x3_r has a value of zero. However, the noise in the control node sometimes makes it jump to a high enough voltage such that the implication becomes erroneously active, forcing the value of v5_2 to an error. In Figure 9, the output of the DC and CDC gates using the 1-node-implication can be less reliable than the output of a normal gate, whereas the error is mitigated by a Schmitt gate due to its biasing behavior. The CDC gate has more noise-immunity than the DC gate because the control signal is connected to two transistors, which will attenuate the noise due to larger gate capacitances. The DC gate using a 2-node-implication has an even higher immunity to the noise in the source node, which is due to large capacitances and logic masking effect.

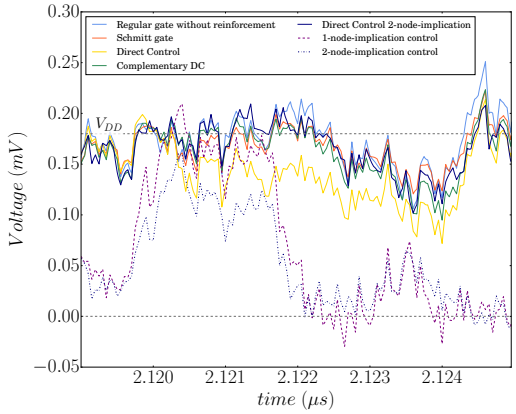

Figure 9: Simulation of v5_2 with inactive implication using different versions of gates.

5.4 Transistor sizing for noise immunity

It is worth noting that increasing the size of a noise-susceptible node is a known technique for improving noise immunity [14], [17]. In our error-correcting gates, the size of the control transistor is critical, since it affects the delay and error-mitigation performance, as well as immunity from noise on the implication control signals. The error-correcting behavior improves as the size of the control transistor increases. However, the output becomes more sensitive to noise on the control signal. That is, while the larger sized transistor can dampen noise due to larger gate capacitance, it will also switch on more easily when there are high levels of noise on the control signal. Sizes of transistors implementing 2-node-implications are larger than 1-node-implications in order to compensate for the series resistance across the 2 control transistors.

As in [17], gate-upsizing can achieve error-mitigation at the cost of area, delay and power. We compared the gate-upsizing method and our implication-based approach in terms of area, delay, power and error mitigation performance. We simulated a double-sized 2-input NOR gate and found the delay to be approximately 4.78ns and the average power dissipation to be around 1.60nW. Compared to the results in Tables 1 and 2, there are no significant differences between a double-sized NOR gate and the DC and CDC gates.

Figure 10 shows a simulation comparing error-mitigation performance between our implication-based approaches and gate up-sizing. Shown is the voltage trace of signal v5_2

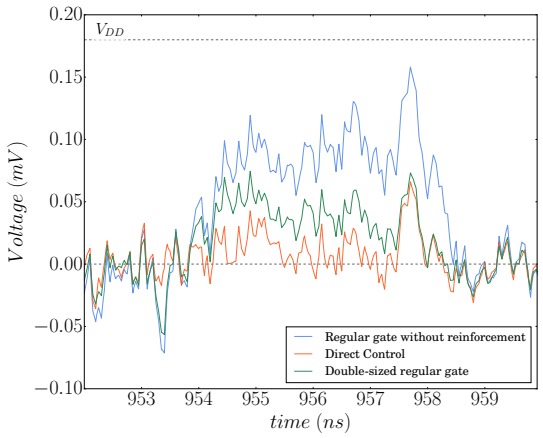

Figure 10: Comparison of error-mitigation behavior between the DC gate and gate-upsizing approach.

from Figure 5. The red trace shows v5_2 when the transistor sizes of gate G4 are doubled, whereas the green trace shows v5_2 when gate G4 is replaced with a DC reinforcement gate implementing the implication [x3_r: 1 → v5_2:0]. Visually, the error-mitigation performance of the DC is better than a regular double-sized NOR gate. An explanation for this behavior is that while gate sizing helps filtering out the noise at the input of the gate by increasing the input capacitance, our reinforcement approach forces the output to the correct logic value. Also, our method is more cost-effective in terms of area for reinforcing a single node.

6. CONCLUSIONS

As circuit operating voltages move to increasingly lower levels, new techniques will be required to provide immunity from various sources of noise. In this work, we propose and analyze two types of logic implications and three version of reinforced logic gates. The 2-node-implication method gives more choices of implications and more tolerance to noise due to error-masking effect and large capacitances. However, this method causes larger delays and power consumption than the 1-node-implication method. The Direct Control (DC) and Complementary Direct Control (CDC) gates are more efficient than the previously proposed Schmitt gate in terms of area, power and delay. Also, the Schmitt gate can only mitigate the error caused by the inputs while the other two versions can also mitigate the error caused by the noise added at the output. However, the DC and CDC gates are more sensitive to the noise in the control signal than the Schmitt gate. Future work will focus on increasing the reliability of the control signal and implication selection. In addition, we will investigate whether our reinforcement technique is useful for mitigating the effects of other types of errors, such as timing errors due to aging effects. We plan to investigate a broader range of signal failures as we continue to analyze the benefits of our approach.

7. ACKNOWLEDGMENTS

This work is supported by NSF under Grant CCF-1525486.

8. REFERENCES

[1] S. Almukhaizim and Y. Makris. Soft error mitigation through selective addition of functionally redundant wires. *IEEE Trans. Rel.*, 57(1):23–31, 2008.

[2] N. Alves, A. Buben, K. Nepal, J. Dworak, and R. I. Bahar. A cost effective approach for online error detection using invariant relationships. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 29(5):788–801, 2010.

[3] N. Alves, Y. Shi, J. Dworak, R. I. Bahar, and K. Nepal. Enhancing online error detection through area-efficient multi-site implications. In *VTS*, pages 241–246, 2011.

[4] M. Baze, S. Buchner, and D. McMorrow. A digital CMOS design technique for SEU hardening. *IEEE Trans. Nucl. Sci.*, 47(6):2603–2608, 2000.

[5] B. L. Dokic. CMOS NAND and NOR schmitt circuits. *Microelectronics J.*, 27(8):757 – 765, 1996.

[6] M. Donato, R. I. Bahar, W. Patterson, and A. Zaslavsky. A simulation framework for analyzing transient effects due to thermal noise in sub-threshold circuits. In *GLSVLSI*, pages 45–50, 2015.

[7] M. Donato, R. I. Bahar, W. Patterson, and A. Zaslavsky. A fast simulator for the analysis of sub-threshold thermal noise transients. In *DAC*, 2016.

[8] M. Donato, F. Cremona, W. Jin, R. I. Bahar, W. Patterson, A. Zaslavsky, and J. Mundy. A noise-immune sub-threshold circuit design based on selective use of Schmitt-trigger logic. In *GLSVLSI*, pages 39–44, 2012.

[9] L. Garcia-Leyva, A. Calomarde, F. Moll, and A. Rubio. Novel redundant logic design for noisy low voltage scenarios. In *LASCAS*, pages 1–4, 2013.

[10] K. Mohanram and N. Touba. Cost-effective approach for reducing soft error failure rate in logic circuits. In *ITC*, volume 1, pages 893–901, 2003.

[11] G. Moore. Cramming more components onto integrated circuits. *Proc. IEEE*, 86(1):82–85, 1998.

[12] K. Nepal, N. Alves, J. Dworak, and R. I. Bahar. Using implications for online error detection. In *ITC*, pages 1–10, 2008.

[13] K. Nepal, R. I. Bahar, J. Mundy, W. Patterson, and A. Zaslavsky. Designing nanoscale logic circuits based on markov random fields. *J. Electron. Test.*, 23(2):255–266, 2007.

[14] R. Rao, D. Blaauw, and D. Sylvester. Soft error reduction in combinational logic using gate resizing and flipflop selection. In *ICCAD*, pages 502–509, 2006.

[15] S. Sinha, G. Yeric, V. Chandra, B. Cline, and Y. Cao. Exploring sub-20nm FinFET design with Predictive Technology Models. In *DAC*, pages 283–288, 2012.

[16] S. Yang. Logic synthesis and optimization benchmarks user guide: Version 3.0. Technical report, MCNC Technical Report, 1991.

[17] Q. Zhou and K. Mohanram. Gate sizing to radiation harden combinational logic. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 25(1):155–166, 2006.