Analysis and Mitigation of Multiple Radiation Induced Errors in Modern Circuits

by

Adam Watkins

M.Sc., Electrical & Computer Engineering, Southern Illinois University, 2012

A Dissertation
Submitted in Partial Fulfillment of the Requirements for the
Master of Science Degree

Department of Electrical and Computer Engineering
in the Graduate School
Southern Illinois University Carbondale
July, 2016

DISSERTATION APPROVAL

TITLE (ANALYSIS AND MITIGATION OF MULTIPLE RADIATION INDUCED ERRORS IN MODERN CIRCUITS)

By

Adam Watkins

A Dissertation Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in the field of Electrical and Computer Engineering

Approved by:

Dr. Spyros Tragoudas , Chair

Dr. Heather Quinn

Dr. Themistoklis Haniotakis

Dr. Haibo Wang

Dr. Shaikh Ahmed

Graduate School
Southern Illinois University Carbondale
(Date of Approval)

AN ABSTRACT OF THE DISSERTATION OF

Adam Watkins, for the Doctor of Philosophy degree in Electrical and Computer Engineering, presented on November 2nd, at Southern Illinois University Carbondale. (Do not use abbreviations.)

TITLE: ANALYSIS AND MITIGATION OF MULTIPLE RADIATION INDUCED ERRORS IN MODERN CIRCUITS

MAJOR PROFESSOR: Dr. S. Tragoudas

Due to technology scaling, the probability of a high energy radiation particle striking multiple transistors has continued to increase. This, in turn has created a need for new circuit designs that are can tolerate multiple simultaneous errors. A common type of error in memory elements is the double node upset (DNU) which has continued to become more common. All existing DNU tolerant designs either suffer from high area and performance overhead, may lose the data stored in the element or are vulnerable to an error after a DNU occurs which makes the devices unsuitable for clock gating. In this dissertation, a novel latch design is proposed in which all nodes are capable for fully recovering their correct value after a single or double node upset which is referred to as DNU robust. The proposed latch offers lower delay, power consumption and area requirements compared to existing DNU robust designs.

Multiple simultaneous radiation induced errors are a current problem that must be studied in combinational logic. Typically, simulators are used early in the design phase which use a netlist and rudimentary information of the process parameters to determine the error rate of a circuit. Existing simulators are able to accurately determine the effects when the problem space is limited to one simultaneous error. However, existing methods do not provide accurate information when multiple concurrent errors occur due to inaccurate approximation of the glitch shape when multiple errors meet at a gate. To improve existing error simulation, a novel analytical methodology to accurate determine the pulse shape when two simultaneous errors occur is proposed. Through extensive simulations, it was shown that the proposed methodology matches closely with HSPICE while providing a speedup of 15X.

The analysis of the soft error rate of a circuit has continued to be a difficult problem due to the calculation of the logical effect on a pulse generated by a radiation particle. The most common existing methods to determine logical effects use either exhaustive input pattern simulation or binary decision diagrams. The problem with both approaches is that simulation of the circuit can intractably time consuming. To solve this issue, a simulation tool is proposed which employs an adaptive partitioning algorithm to reduce the simulation time and space overheads of binary decision diagram based simulation. Compared to existing simulation tools, the proposed tool can simulate larger circuits faster.

DEDICATION

This dissertation is dedicated to my parent Daniel and Debra Burris and my wife Yijing Zhang Watkins for their support during the creation of this work.

ACKNOWLEDGMENTS

(NOT REQUIRED IN RESEARCH PAPER)

I would like to thank Dr. Jones for his invaluable assistance and insights leading to the writing of this paper. My sincere thanks also goes to the seventeen members of my graduate committee for their patience and understanding during the nine years of effort that went into the production of this paper.

A special thanks also to Howard Anton, from whose book many of the examples used in this sample research paper have been quoted. Another special thanks to Prof. Ronald Grimmer who provided the previous thesis template upon which much of this is based and for help with graphics packages.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The continued reduction of the transistor feature size has led to the increase in the vulnerability of modern circuits to error. This effect, in turn, has made modern circuits more susceptible to radiation induced errors in space and terrestrial environments. A radiation induced error, commonly referred to as a soft error, occurs when a high energy particle from space or packaging strikes a transistor. Shown in Fig. 1, the particle deposits energy in the active volume generating electron-hole pairs. This creates a new diffusion region that could allow a non-conducting device to temporarily conduct current. The mechanism causes a temporary voltage pulse, referred to as a single event transient (SET) to occur at the device.

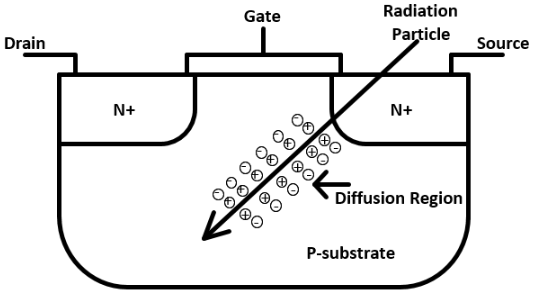


Figure 1: A energetic particle creating electron-hole pairs in a transistor.

In the case of a radiation particle striking a memory storage element, the data stored in the device can be changed. If the device loses its value due to a strike by a single particle, this is referred to as a single event upset (SEU). Additionally, due to the constant scaling down of the feature size, a single particle can also strike two transistors simultaneously which is referred to as a double node upset (DNU). Fig. 2 gives a diagram of a SEU tolerant latch to demonstrate the DNU phenomenon. In the diagram, radiation is denoted as a high energy particle which passes through two transistors on a DICE latch [2]. In the case of a SEU, the DICE latch would normally be able to tolerate the error. As can be observed, the particle passes through two transistors causing the respective node to switch from ”1” on the first node to ”0” and from ”0” to ”1” on the second node. The upset of both nodes drives the remaining two nodes to an erroneous value. This observance is alarming since it signals that current designs are not sufficient for future processes. In turn, this provides a need for new memory element designs that can tolerate DNUs.

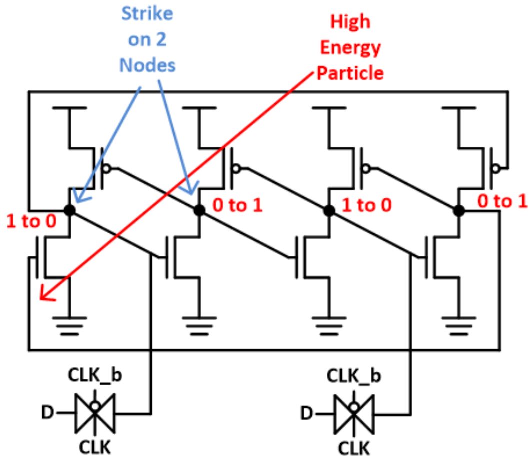


Figure 2: A particle striking two node concurrently causing a double node upset.

Additionally, modern circuit designs employ a technique referred to as clock gating. Clock gating is defined as setting the clock to constant value thus reducing the power consumed. Use of clock gating may lead to cases where the current state held by the memory element must be held for many clock cycles. This need leads to a much longer than normal vulnerable window in which an improperly hardened device may lose the stored value. While there are existing DNU tolerant designs [6, 7, 19], these designs move to a high impedance state after a DNU. If a DNU occurs while the latch is gated, the voltages within the latch may degrade. In Chapter 1, we aim to alleviate this problem by providing an efficient design that is capable of recovering all nodes after a DNU occurs. This, in effect, guarantees that the data will be held even if the latch is struck by a DNU while gated.

In addition to the effects on memory elements, accurate approximation of the error rate of a circuit is also important. The goal of circuit simulation is the accurate estimation of the soft error rate (SER). In this work, the focus is on the evaluation of combinational circuits. Typically, this type of simulation entails the estimation of the resulting pulse shape when a particle strike a transistor, the determination of the boolean functions that allow the pulse to propagate and the estimation of the latching characteristics at an output flip-flop. Using these parameters, the probability of the pulse reaching an output flip-flop is determined. Assume that the probability of an error reaching and being latched in a flip-flop at output i of the circuit is represented as $P(O_i)$, the particle hit rate in a particular area is given as R_{PH}, the fraction of particle hits resulting in charge generation as R_{eff} and A_{cir} gives the area of the circuit. The equation for the SER at output O_i is given below [10].

$$SER_{O_i} = P(O_i) * R_{eff} * R_{PH} * A_{cir} \tag{1}$$

The focus of all SER estimation simulators is the calculation of the term $P(O_i)$. This is a difficult problem in combination circuits due to the presence of the following three masking factors: electrical masking, logical masking and temporal masking. Electrical masking deals with the calculation of the pulse shape as it propagates through the circuit. Logical masking is the estimation of the logical 1's and 0's and how they may mask the pulse. For example, if a NAND gate has a value of "0" on an input, the pulse will not propagate since the output will be held to a "1". Lastly, temporal masking involves the latching characteristics, specifically the set-up and hold times, of the output flip flop. Accurate consideration of all three masking effects is crucial to accurate SER estimation.

Accurate consideration of the electrical masking effect is an important but often simplified component of SER estimation. The most common method to calculate the pulse shape, proposed in [13] uses a linear line to approximate the rising and falling transitions giving a trapezoidal shape. While this method executes quickly and is easy to implement, it does not provide an accurate estimation of the pulse shape. As can be observed in Fig. [ref], a transient pulse does not take a trapezoidal shape in a real case. For this reason, accurate consideration of the pulse shape must consider the non-linear aspects of the pulse. In [18], the authors proposed an enhanced pulse approximation method which is accurate within 5% of HSPICE. However, their method has the drawback that it can only consider a single pulse arriving at a gate. In the case of multiple transient pulses injected into a circuit, referred to as a multiple event transient (MET), the likelihood of two or more pulses arriving at a gate simultaneously as shown in Fig. 4 is substantially increased. To accommodate this trend, future soft error simulators must accurately consider this effect.

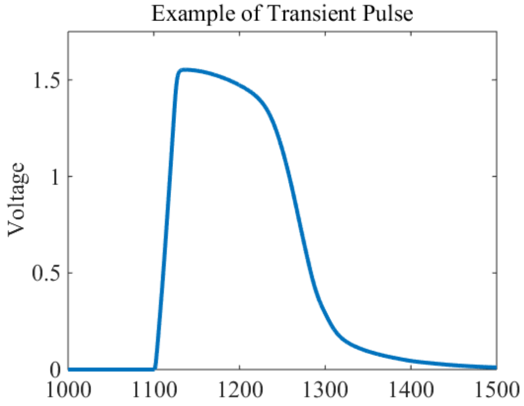


Figure 3: An example of a radiation induced transient pulse.

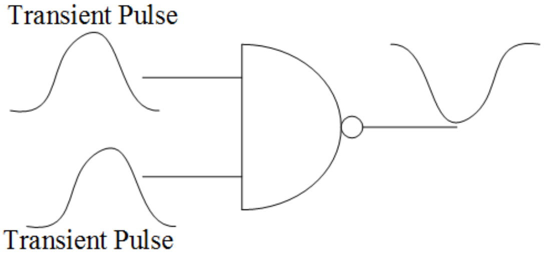


Figure 4: Two pulses arriving at a gate input.

In addition to the electrical masking effect, the logical masking effect must also be accurately considered. Many existing methods use Monte Carlo based simulation which consists of either exhaustively apply all patterns or randomly applying a subset of patterns. [18, 17, 11, 22, 16] While these methods are able to simulate with a very low memory overhead, they have an intractable execution time for circuits with more than 30 inputs. This problem is further compounded by the fact that due to the nearly infinite number of possible pulse durations and strike locations. To alleviate this issue, the authors in [23] propose the use of probabilistic transfer matricies [PTM]. While the matricies do provide exact calculation of the logical masking effect, their memory usage blows up even for relatively small circuits. As an improvement to the use of PTMs, the use of decision diagrams have become more common. [20, 9] Decision diagrams offer an improved approach since they balance the cost of execution time and memory consumption. However, decision diagrams are not a perfect solutions since, like PTMs, blow up if the circuit is sufficiently large. To solve this issue, the authors in [20] propose the use of partitioning to limit the size of the decision diagrams. The drawback with this approach is that each time the circuit is partitioned, additional error is added to the result. To ensure that the effect on the result is low, the number of partitions should be minimized based on the system used and amount of time allowed.

In this dissertation, novel approaches to the aforementioned problems are proposed. In Chapter 1, a latch design is proposed which is capable of recovering all nodes after a DNU. This design has specific applications in circuits that employ clock gating. Chapter 2 provides an enhanced analytical pulse approximation algorithm which can determine the output pulse shape when two pulses arrive at a gate input. The method is implemented and compared to existing algorithms and HSPICE. Chapter 3 discusses a soft error simulator which adaptively partitions the circuit based on the size of the BDD functions. Results are provided which compare the effect of the partitioning algorithm on the calculation of the SER.

CHAPTER 1

ROBUST LATCH DESIGNS FOR MULTIPLE NODE UPSETS

As the transistor feature side continuously scales down to improve performance, modern circuitry continues to become more susceptible to radiation induced errors commonly referred to as a soft error. Soft errors can manifest from either neutron particles originating from space or alpha particles from packaging. A soft error occurs when an energetic particle hits the diffusion region of a reverse bias transistor. This, in turn, allows an "off" transistor to temporarily conduct current which can cause a voltage change in a node connected to the affected transistor. If the error occurs in combinational logic, the resulting voltage pulse may be stored in a connected flip flop thus causing an error. On the other hand, if the error occurs in memory or a latch during the hold phase, the stored data may change. To mitigate this effect, there is a need for design methodologies that reduce the vulnerability of circuitry to radiation effects.

In this chapter, the focus is on the reliability of latches. There has been extensive research in the field of hardening latches against single even upsets (SEU). The simplest and most common design in safety critical applications is the triple modular redundancy (TMR) latch. This design consists of 3 standard latches connected to a 3-input majority voting circuit. While this design is tolerant against errors, it has the drawback of high area, delay and power consumption. For this reason there have been many other designs proposed that offer high SEU reliability with lower area, delay and power consumption. The first and most common cell is the DICE cell proposed in [2]. The design in [2] consists of eight cross-coupled PMOS and NMOS transistors connected in series which forms four nodes. Due to the relatively high delay and power consumption of the DICE latch, there have been many other SEU tolerant latch designs proposed that provide reliability using blocking Muller C-elements, redundancy or delay in the feedback path [14, 4, 5, 15, 8, 21, 12].

In more recent times, the further reduction of the transistor feature size has increase the likelihood of a single event causing a transient on multiple nodes simultaneously, commonly referred to as a single event multiple upset (SEMU). This trend necessitates the development of new latch designs that are tolerant to multiple node strikes to guarantee reliability in current and future technologies. As in the SEU case, the goal of these designs are to minimize the power, delay and area overheads. However, contrary to the SEU case, the latches are designed to tolerate two simultaneous errors, commonly referred to as a double node upset (DNU). Currently there are many existing latch designs that are tolerant to DNUs which are discussed in Section 1.1.

Many modern circuit designs employ a technique commonly referred to as clock gating to further reduce the power consumption. Clock gating consists of setting the clock to a stable value or "gating" the clock. If clock gating is used with a latch, it may need to hold the current state for many clock cycles. In the presence of DNUs, this increases the likelihood of multiple errors occurring during the hold phase. In many existing DNU tolerant designs, a DNU puts the latch to a high impedance state in which the correct value could be lost if the latch experiences a further SEU or DNU before the transparent mode. Additionally, in many of these designs, a DNU moves the output to a high impedance state which implies that the data could discharge if the latch is gated for a sufficient number of cycles. For this reason, there is a need for new designs that are capable of holding the correct output value after a DNU for any number of clock cycles. In the chapter, we classify all DNU tolerant designs as either DNU robust or DNU non-robust. A DNU robust design is defined as being capable of resisting further errors and by not allowing any high impedance states after a DNU occurs. A DNU non-robust design is a latch that does not meet the all of stated criteria. In this chapter, a novel latch design referred to as the HRDNUT (Highly Robust Double Node Upset Tolerant) is proposed.

1.1 BACKGROUND ON EXISTING DNU TOLERANT LATCHES

Currently, there are a few existing DNU tolerant designs. The first proposed design found in [7], referred to as the DNCS latch, consists of two DICE cells connected to an output Muller C-element. This design tolerates DNU's since each DICE element requires a DNU to flip its state. Since the assumption is that only two errors can occur at once, in the worst case only one DICE element flips its state. Due to the C-element, the latch output does not change value. This design has been shown to be very resilient to DNUs at a very high cost of area, delay and power. The authors in [6] propose an enhanced design compared to [7]. Their latch design consists of six 2 input C-elements connected in series which are then fed into a 3 input C-element. Like the DNCS latch, this design offers high resiliency to DNUs, however the power consumption and area overheads are still very high.

More recently, a highly area and power efficient design has been proposed in [19] and is referred to as the HSMUF latch. Fig. 1.1 provides the design. The HSMUF uses the TP-DICE [1] structure which consists of 6 cross-coupled elements. In the case of a DNU, if the error is on an adjacent node (such as a strike on n1 and n2), the TP-DICE element will fully recover the previous state. However, if the strike occurs on two non-adjacent nodes, the TP-DICE will not fully recover leaving one output node with an erroneous value, one node at high impedance and the remaining output node held at the error-free value. To provide reliability, the three nodes are connected to a C-element, as in Fig. 1.1, which allows the correct value to be held at the latch output.

While all of the previously discussed designs do provide high DNU reliability, none of them are classified as DNU robust since a DNU will result in high impedance states on the internal and output nodes. If an error occurs after a DNU, these latch designs will flip their held value. A popular remedy to this issue is to place a weak keeper on the latch output as in Fig. 1.1. However, adding a weak keeper greatly increases the power, area and delay overheads since the output C-element must be re-sized so that the C-element's driving strength exceeds that of the keeper. According to our simulations in Section 1.3, the addition of the keeper to the HSMUF latch nearly triples the power consumption and delay. Additionally, the latch is still vulnerable to error after a DNU since the TP-DICE is in a high impedance state.

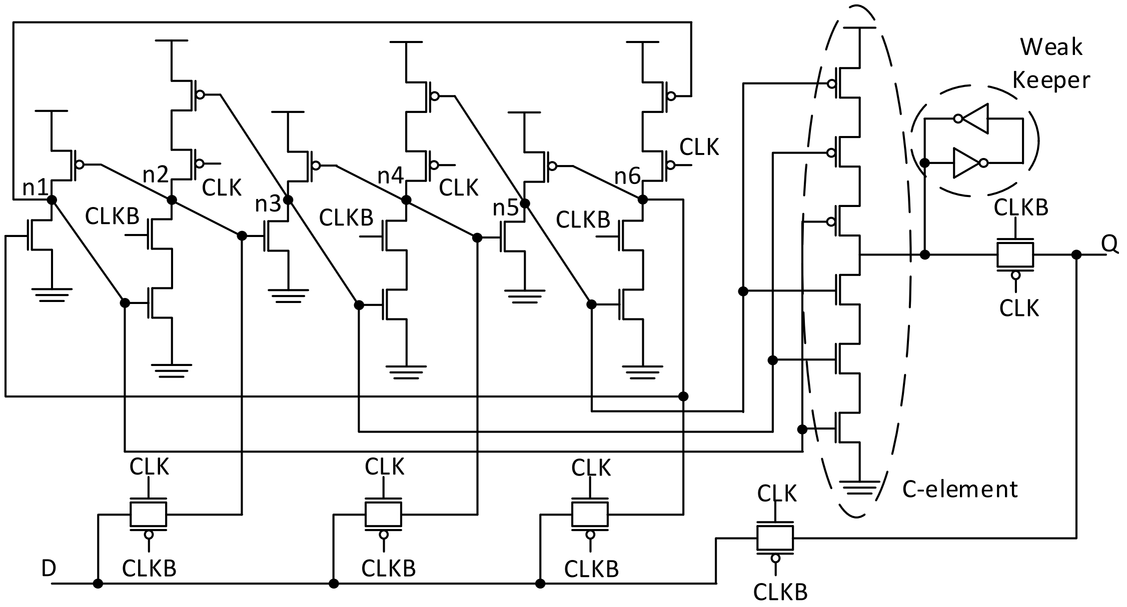


Figure 1.1: HSMUF latch [19] with a weak keeper on the output.

To the author's knowledge, the existing most efficient DNU robust design capable of recovering all nodes after a DNU is the DONUT latch [3] in Fig 1.2. The design, as proposed in their paper, uses only 36 transistors but has a much higher power consumption compared to the HSMUF (See Section 1.3). The reason for the high power consumption is due to contention on the input lines during the transparent mode. For example, if we observe node *n2* in Fig. 1.2 during the transparent mode, the node is driven by three cross-coupled elements. This contention will increase the amount of time required to change the node thus drastically increasing the dynamic power consumption. To optimize their design, we create the 48 transistor DONUT-M latch in which each component connected to an input node is modified, as shown in Fig. 1.3 so that the line is at high impedance for the whole duration of the transparent mode. This, in effect, removes the data contention problem thus reducing the overall dynamic power and delay.

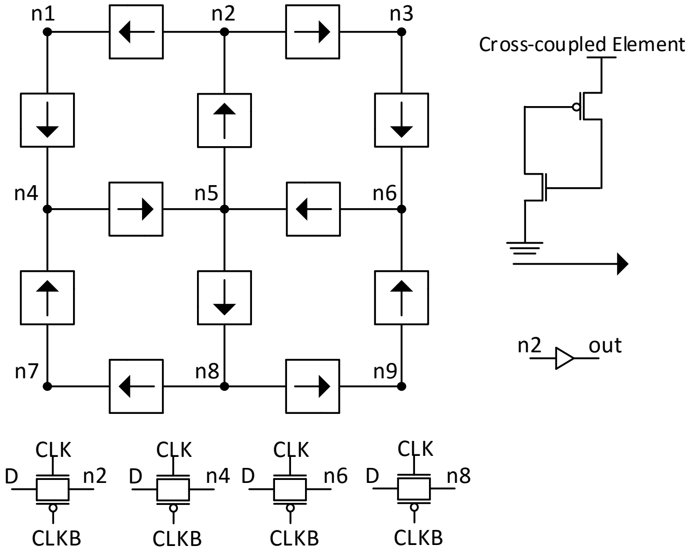


Figure 1.2: DONUT latch as proposed in [3].

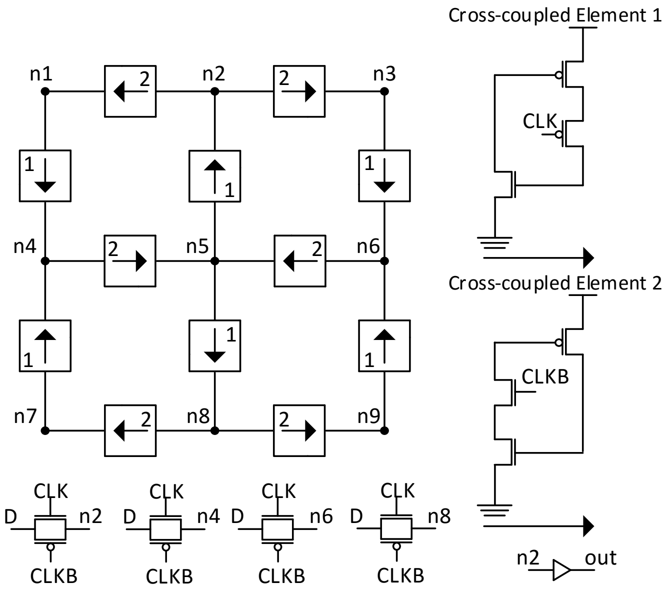


Figure 1.3: Modified low-power DONUT latch.

1.2 PROPOSED HRDNUT LATCH DESIGN

In this section we discuss the proposed DNU robust latch. The latch implementation is based on three cross connected storage loops connected to three C-elements. The basic design of the storage loop is given in Fig. 1.4. The data loop is based on the standard latch design with a 3-input C-element inserted to replace one of the inverters. The purpose of the C-element is to separate the feedback loop so that an error will not be held. Additionally, a PMOS is connected to the positive clock signal (CLK) and a NMOS is connected to the negative clock signal (CLKB) to remove contention when data is loaded to the latch. As in the modified DONUT latch, the addition of these transistors drastically reduces the delay and power consumption.

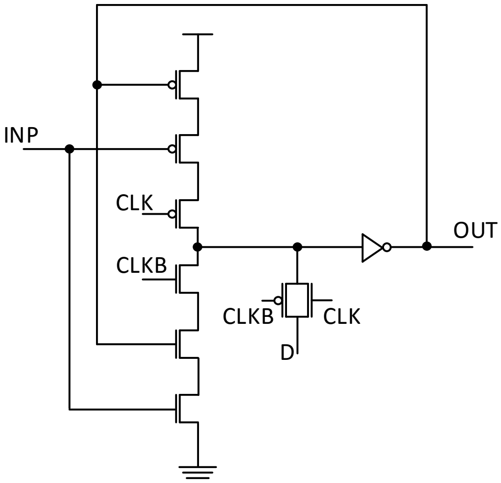


Figure 1.4: Basic data storage loop block.

Using the basic storage block we construct the block based latch as in Fig. 1.5. The latch was designed with the goal of ensuring that none of the nodes directly drove itself. For example, it can see that in Fig. 1.4 the node *out* is fed into the input of the 3-input C-element. If an error strikes node *out*, the cell will never be able to recover its previous state since one of the C-element inputs will be held to an erroneous value by its output. To prevent this issue, our design is based on cross-connecting three of the storage loop blocks so that the C-element is driven by three separate block outputs. In Fig. 1.5 we provide a basic latch design using this idea. If a single error occurs on any node in this design, the circuit is capable fully recovering the previous data.

To demonstrate this, consider a strike on node *n2*. When the strike occurs, the erroneous value will be propagated to the C-elements driving nodes *n1* and *n3*. However, since there is no change on *n1* or *n3*, the C-elements *C1* and *C3* will hold their previous value thus preventing the error from propagating to the output. Additionally, since node *n2* is driven by nodes *n1* and *n3*, *n2* will completely recover the correct state.

A problem, however, with the latch in Fig. 1.5 is that it is not capable of tolerating DNUs. For example, if an error occurs on nodes *n1* and *n2* the erroneous values will propagate to the inputs of C-element *C3* and flip the value of *n3* thus changing the output value. However, since the latch has recovery capability for SEUs, we modify it so it can tolerate DNUs and recover all nodes to the previous state. In Fig. 1.7 we provide the schematic of the proposed HRDNUT latch. The design uses the block-based latch in Fig. 1.5 as a base and adds additional C-elements to prevent errors from being held by the data loop.

Initially, we will evaluate the HRDNUT latch during normal operation. When the positive clock signal (CLK) has a high value and the negative clock signal (CLKB) has a low value, the latch is in transparent mode. At this stage, the transistors connected to the clock signal in C-element *C1* deactivates the PMOS and NMOS stacks thus causing the node *n1* to be in a high impedance state. This, in effect, reduces data contention thus reducing delay and dynamic power consumption. Next, the data is loaded through the pass gates connected to nodes *n1*, *n22* and *out*. Since the output node *out* is loaded directly, the data to out delay is minimized and all nodes are set to their respective error free values. When CLK changes to a low value and CLKB to a high value, the latch moves into the hold mode. In this stage, the pass gates are deactivated and the state of the latch is held since each node is driven to the correct value using a C-element. Fig. 1.6 provides

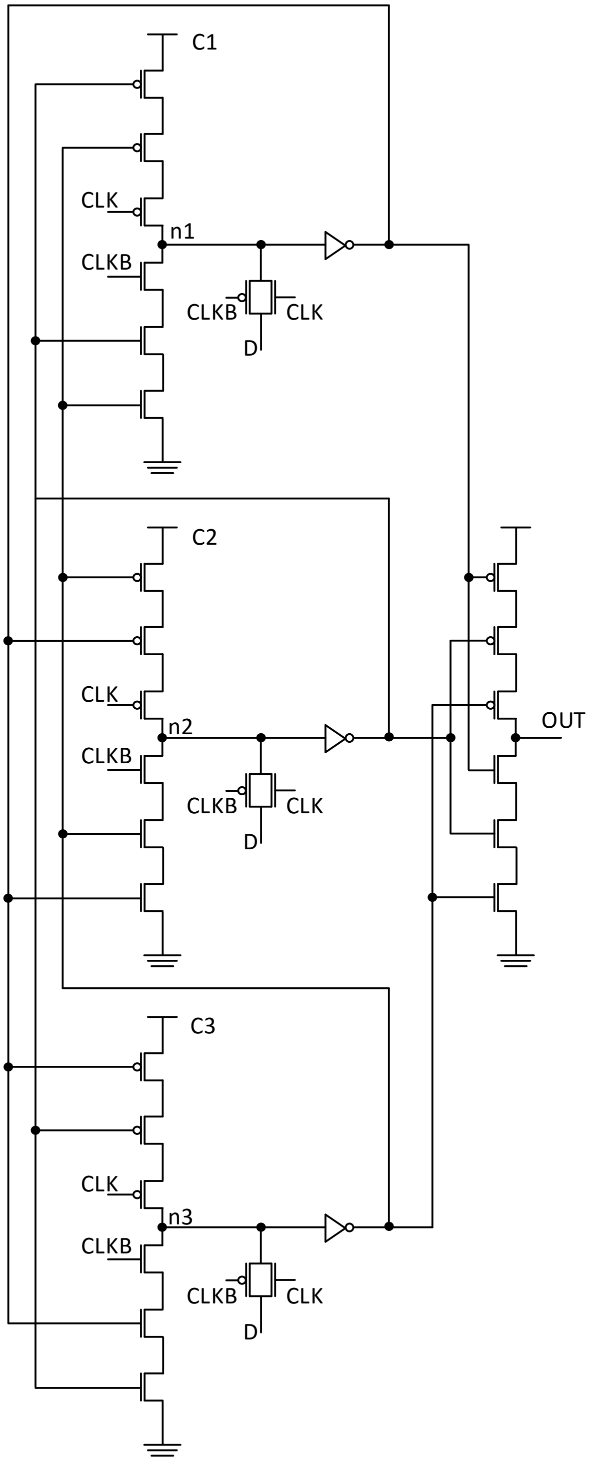


Figure 1.5: Schematic of the block-based latch.

the waveforms of the CLK, D and OUT nodes for both the transparent and hold modes of operation.

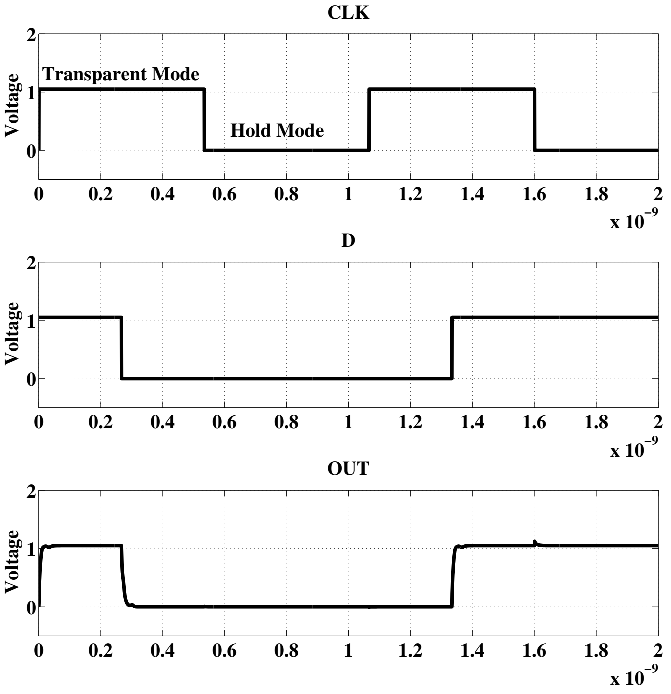


Figure 1.6: Waveforms of the HRDNUT latch during normal operation.

In the case of an SEU, the HRDNUT retains the excellent resiliency of the block based latch and the ability to recover every node after an error. In the case of any internal node being struck by an error, the latch will not change value due to all internal C-elements requiring at least 2 identical input values to change values. In the case of an error hitting the output node *out*, the latch fully recovers since *out* does not directly drive C-element *C7*.

Lastly, the latch will be evaluated in the case of a DNU. Note that unless otherwise stated, it is assumed that the analysis applies to both when D=0 and D=1. For the analysis, the possible DNU strike combinations are categorized into 9 distinct cases based on their effect in the HRDNUT latch. The categories are discussed in detail below:

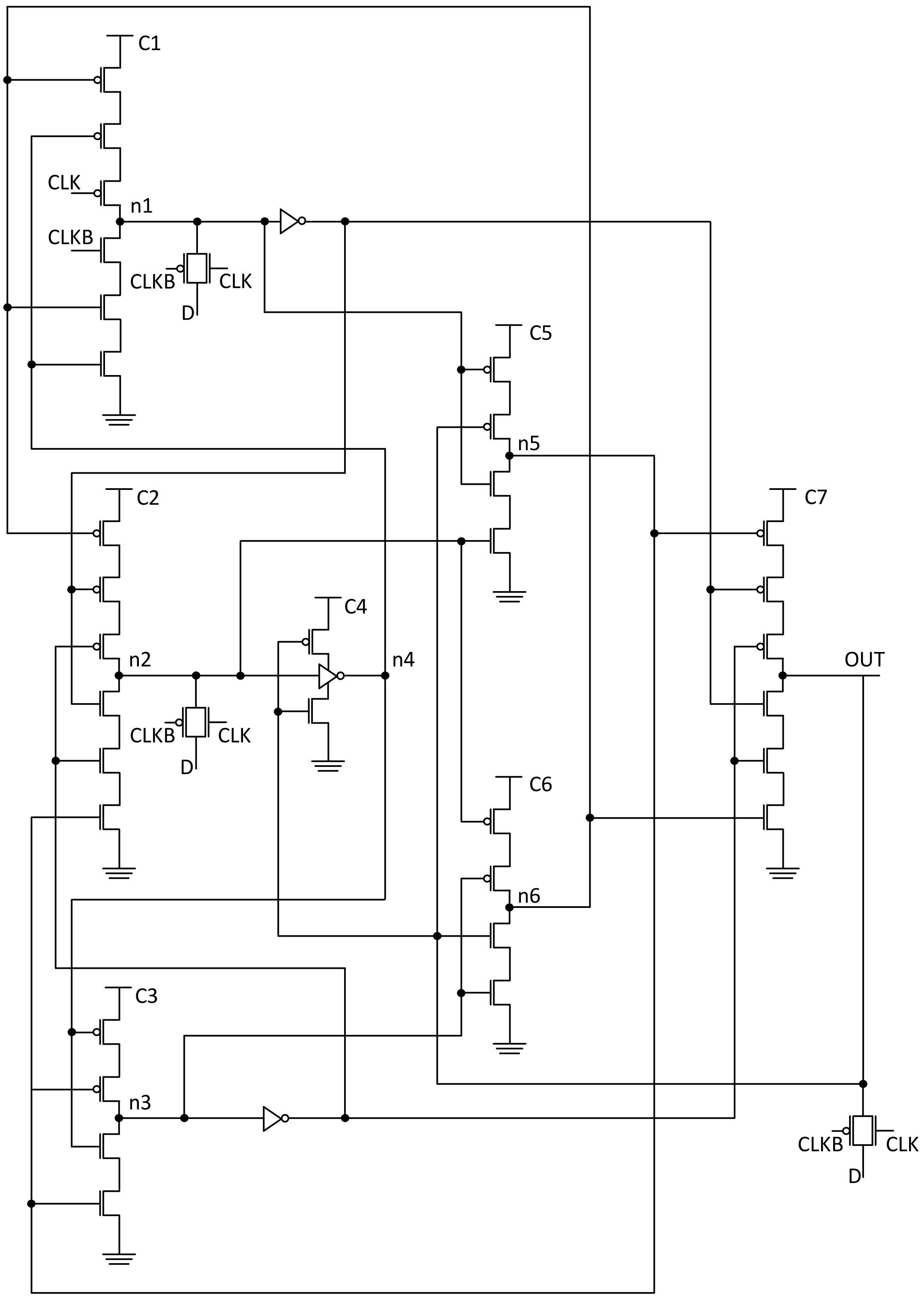


Figure 1.7: Schematic of the HRDNUT latch.

1. Consider strikes at nodes *n1* and *n2*. In this case, the error at *n1* will propagate to C-elements *C5* and *C7* but will not cause a flip since the error at *n2* will be blocked by C-element *C4*. Additionally, since the inputs of C-elements *C1* and *C2* are unchanged, the nodes will recover their initial values. This analysis can be applied to node combinations containing node *n2* except for the combination with node *out* since the error will be blocked by C-element *C4*.

2. In the case of a DNU upsetting nodes *n2* and *out*, the error at *n2* will propagate through C-element *C4*. However, C-elements *C1* and *C3* will block the error and nodes *n1*, *n3*, *n5* and *n6* will hold their values thus driving node *out* to the correct state.

3. Consider when a DNU strikes nodes *n1* and *n5*. In this case, the error at *n1* hits the output of C-element *C1* which is propagated to *C7*. The error on *n5* is also propagated to C-element *C7*. Since node *n3* and the inputs of C-elements *C1* and *C5* are unaffected by an error, the output retains the error-free value and the latch fully recovers the previous state. The above analysis also applies to the node combination (*n3*, *n6*).

4. In the case of a DNU hitting nodes *n3* and *n4*, the error at *n4* is propagated to C-element *C3* and the error at *n3* is propagated to *C7* and *C6*. After the error on *n3* subsides, *C4* will drive node *n4* and, due to the connection at *C3*, node *n3* back to the error-free value. The node combination (*n1*, *n1*) can be analyzed similarly. For the node combinations of (*n4*, *n5*) and (*n4*, *n6*), the latch will also recover the previous result since the inputs to *C4* are unchanged. This implies that after the error occurs at *n4*, the node will be driven back to the correct value thus also driving the nodes *n5* or *n6* back to the correct value.

5. When a DNU upsets the combination of *n4* and *out*, the error at *out* is propagated

to *C4*, *C5* and *C6* and the error at *n4* to *C1* and *C3*. Since none of the inputs to *C7* are changed by the error, *out* is flipped back to its error-free value which drives *n4* through *C4* back to its previous state.

6. Consider when a DNU strikes nodes *n1* and *n3* being struck. In this case, the errors are propagated to C-elements *C2*, *C5*, *C6* and *C7*. However, since the errors do not manifest into an error on any other node, the latch fully recovers from the error.

7. When a DNU strikes the nodes *n1* and *n6*. The error at node *n6* propagates to *C1* and *C7* while the error at *n1* also propagates to *C7*. Due to the error-free node *n3* driving *C7*, the previous value is held at the output by *C7*. Additionally, *n3* will drive *C6* back to its previous value thus driving *C1* back to the error free state. This analysis can be applied similarly to the node combination of (*n3*, *n5*).

8. In the case where a DNU strikes nodes *n5* and *out* the error at *n5* propagates to *C7*, *C2* and *C3* and the error at *out* goes to *C4*, a PMOS in *C5* and a NMOS in *C6*. When the error-free value at *out* is 1, the value at *n5* is 0. The error at the nodes change the values to 0 and 1 respectively and the erroneous value at *out* is propagated to the PMOS at *C5* and the NMOS at *C6*. This, in effect, causes the PMOS at *C5* to be activated and the NMOS at *C6* to be deactivated. However, since nodes *n1* and *n2* remain error-free, the NMOS stack of *C5* will drive *n5* back to the correct value. This, in turn, forces *C7* to also drive *out* back to the error-free value. In the case where *out* has an ideal value of 0, the error will be fully recovered since the NMOS stack will be entirely driven by fault-free nodes. The above analysis can be applied to the node combination of (*n6*, *out*).

9. Lastly, we analyze the node combinations (*n1*, *out*), (*n3*, *out*) and (*n5*, *n6*). In these cases the errors do not cause a change on the inputs of any C-elements driving the node thus the previous value will always be recovered.

1.3 SIMULATION RESULTS

The proposed HRDNUT latch has been implement using the 1.05V 32nm PTM library [23] and simulated in HSPICE. All transistors were set to the minimum size with the PMOS widths set to W=80nm and the NMOS widths set to W=40nm. To evaluate the DNU reliability of the design, current pulses were injected for every possible error combination. The injection current was calculated using the equation found in [24]. The equation is given below with τ as the technology dependent constant, Q_o as the injection current value and t as the variable for time.

$$I(t) = \frac{2Q_o}{\tau\sqrt{\pi}}\sqrt{\frac{t}{\tau}}e^{\frac{-t}{\tau}} \tag{1.1}$$

Using equation (1.1) τ was set to 32×10^{-12} and Q_o was set to 5fC. In all simulations, the latch was operated at a frequency of 1Ghz. In Appendix I Figs. 3.5-3.14, the waveforms for each case discussed in Section 1.2 are presented and show that the HRDNUT is fully capable of recovering all nodes in the presence of a DNU.

Next, we compare the HRDNUT to existing SEU and DNU tolerant methods. As in the HRDNUT latch, all latches were designed using the 32nm PTM library and operated at 1Ghz. For the analysis, we compare to the following SEU tolerant latches: DICE [2], FERST [4] and HIPER [14]. Additionally, we also compare to the following DNU tolerant designs: DNCS [7], Interception [6], HSMUF [19] and DONUT [3]. All transistors for the implemented latches were set to minimum width and length except for the designs that use a C-element with a weak keeper. In these designs the C-element's PMOS width was set to W=320nm and the NMOS width was set to W=160nm and the weak keeper was sized to be at minimum width. The C-element was sized so that the output driving strength did not allow the keeper to drive an erroneous value in the event of an error.

To provide a fair comparison, we measure the propagation delay, average power consumption and area of all designs and categorize them base on whether they can tolerate a DNU and if they are robust from error after a DNU occurs. The delay was measured as the time between when a transition occurs on input D to when a transition was observed on the output. The average power was computed using the error-free operation for each latch for a duration of 200ns. To compare the area overhead, we adopt the unit size transistor (UST) metric as in [7] which represents the number of unit sized (minimum width is W=40nm in this case) transistors required for the design. Table 1.1 provides the results of these simulations.

Table 1.1: SPICE Simulations of Existing Latches using the 1.05V 32nm PTM library

Latch	DNU Immune	DNU Robust	Power (μW)	Delay (ps)	Area (UST)
DICE	No	No	1.332	8.145	16
FERST	No	No	3.178	31.648	60
HIPER	No	No	1.292	2.221	27
DNCS	Yes	No	4.948	22.486	61
[6]	Yes	No	5.606	79.168	89
HSMUF	Yes	No	1.871	1.0626	51
HSMUF (Keeper)	Yes	No	3.787	3.945	78
DONUT [3]	Yes	Yes	4.021	14.722	54
DONUT-M (Section 1.1)	Yes	Yes	2.760	8.421	72
HRDNUT (Proposed)	Yes	Yes	2.450	2.310	66

According to Table 1.1 the only DNU robust designs are the two DONUT latch implementations and the HRDNUT. Compared to the modified DONUT latch, the HRDNUT provides DNU robustness while reducing the power consumption and the number of transistors by 11.3% and 8.33% respectively while also reducing the delay by 72.5%. For the above reasons, the HRDNUT is the best design for clock gating applications due to its high robustness, even after a DNU occurs, and lower power, delay and area overheads.

1.4 FILTERING OF TRANSIENTS ARRIVING ON THE LATCH INPUT

In addition to radiation particles hitting the internal nodes in a latch, transient voltage pulses may arrive on the inputs. In a typical digital system design, combinational logic is placed between two sets of flip-flops, one set driving the input and the other being driven by the output of the combinational circuit. While there are masking factors in these types of circuits, it may still be likely that a pulse will arrive at the output of the circuit. A method to filter incoming pulses was proposed in [4]. Their design consisted of routing a single combinational output through two paths, one with no delay and one will sufficient delay that the two pulses will not overlap. The general design is given below.

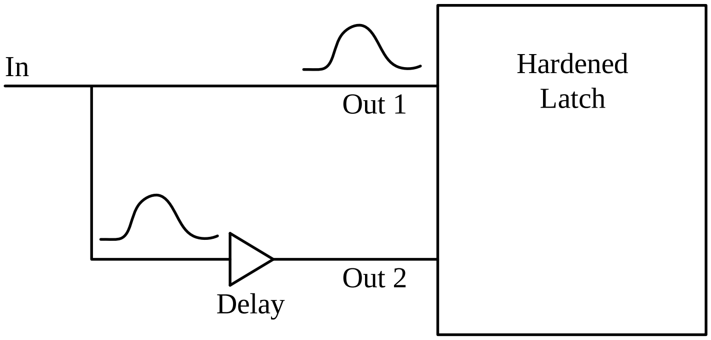


Figure 1.8: The pulse filtering circuit provided in [4].

An issue with the approach in Fig. 1.8 is that in a high radiation environment, a high energy particle can hit the delay element causing a voltage pulse on the node. If this occurs while a pulse being propagated through the circuit, it could delay the pulse at the delay element causing the erroneous value to be latched in the flip flop. A simple way to mitigate this effect is to add an additional delay element with twice the delay. This, in effect, will allow the device to function properly even if an error occurs during the filtering stage. The design of the device is given in Fig. [ref]. Due to the use of three outputs, the device can be used with the HRDNUT latch without modification.

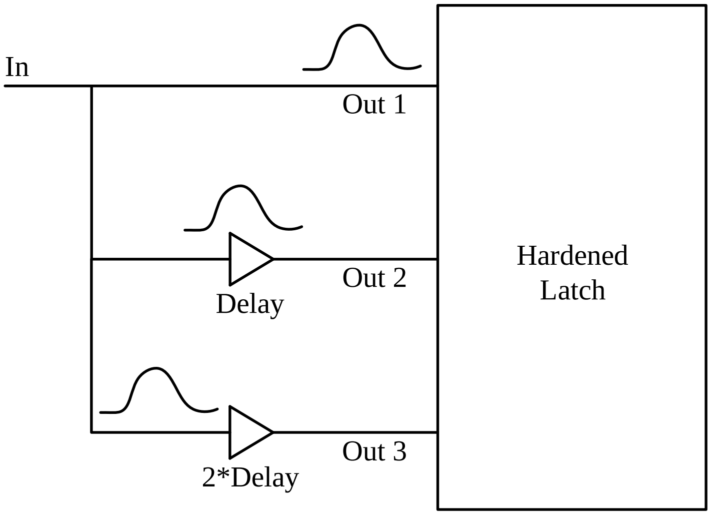


Figure 1.9: The enhanced pulse filtering circuit that can tolerate an error.

1.5 ANALYSIS OF A TRIPLE NODE UPSET TOLERANT LATCH

An inevitable issue with the development of more complex latches to tolerate multiple errors is the increase of the number of transistors and area. This issue leads to the possibility of new latch designs being vulnerable to more simultaneous error such as a triple node upset (TNU). The goal of this section is to estimate the area and device overhead and to evaluate the feasibility of a TNU tolerant latch. To begin the analysis, the design methodology used in the development of the HRDNUT is applied. The basic element used in this design is the block based latch given in Fig. 1.5. In the case of the HRDNUT, 3 blocks were needed to harden the latch against 2 errors. Based on this observance, it is estimated that hardening against TNUs will require 4 blocks. Additionally, each c-element will have and additional transistor added to accompany the extra block. The block based latch for TNUs is given in Fig. 1.10 and shows that TNU hardening requires a % transistor overhead.

It can be extrapolated from the overhead of the block based latch that the overhead of the TNU tolerant latch will be similar. Due to the high overhead, it can be concluded that creating latches that tolerate more than the DNU will be intractably large since a higher area increases the number of multi-node upsets. While it is likely that the TNU tolerant latch will offer more reliability since a TNU should be a more rare case compared to a DNU, the power and area overhead will not offset the increased reliability.

1.6 CONCLUSION

In this section the HRDNUT latch was proposed which is suited for clock gating schemes. Since clock gating may require the latch to remain in a hold state for many clock cycles, the susceptibility of error increases. In many existing designs, a DNU may either change the state of the latch or push the latch into a state were the output may discharge over time due to a high impedance state. A common method to solve this problem is the addition of a weak keeper on the output. As shown in this paper, the addition of the keeper causes much higher power consumption. Since the HRDNUT latch does not stay in a high impedance state after a DNU, the HRDNUT provides high reliability during the whole duration of the hold mode while providing the lowest delay, power and area compared to other latches suitable for clock gating. Simulation results show that the HRDNUT is 11.3% more power efficient while requiring 8% less transistors and 72.5% less delay compared to the highly robust DONUT latch.

Additionally, an enhanced circuit to filter incoming transient pulses was proposed. The new circuit is able to tolerate a single error during the filtering phase. The area additional area of the new filter circuit is dependent on the size of the expected transient but overall is low. Lastly, an analysis of the TNU was conducted and the block based circuit was given. It was determined that a TNU tolerant circuit would have a % greater overhead compared

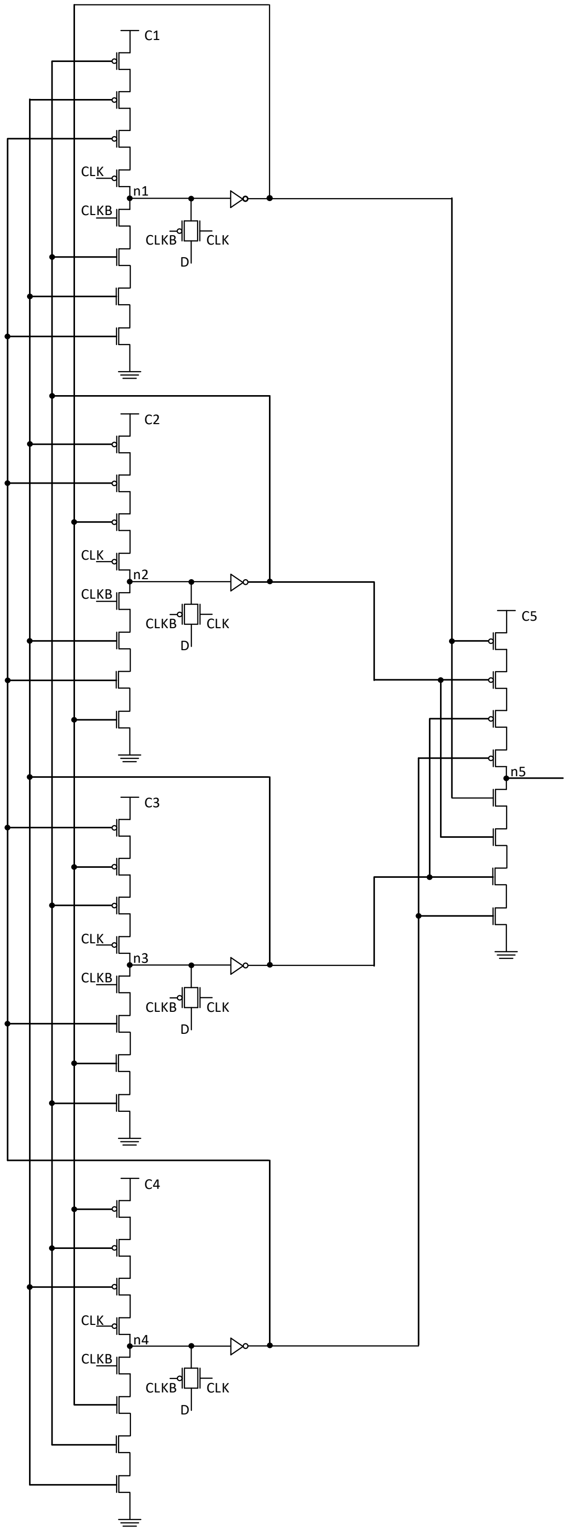


Figure 1.10: Block based latch that forms the basis for a TNU tolerant design.

to the HRDNUT. Based off this finding, it is recommended that a TNU tolerant latch is used in extreme radiation environments where system stability is crucial.

CHAPTER 2

SYSTEMS OF LINEAR EQUATIONS AND MATRICES

2.1 INTRODUCTIONS TO SYSTEMS OF LINEAR EQUATIONS

In this section we introduce base terminology and discuss a method for solving systems of linear equations.

A line in the xy-plain can be represented algebraically by an equation of the form

$$a_1x + a_2y = b$$

An equation of this kind is called a linear equation in the variables x and y. More generally, we define a linear equation in the n variables $x_1, \ldots, x_n$ to be one that can be expressed in the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b \tag{2.1}$$

where $a_1, a_2, \ldots a_n$ and b are real constants.

Definition. A finite set of linear equations in the variables $x_1, x_2, \ldots, x_n$ is called a *system of linear equations.*

Not all systems of linear equations has solutions. A system of equations that has no solution is said to be *inconsistent.* If there is at least one solution, it is called *consistent.* To illustrate the possibilities that can occur in solving systems of linear equations, consider a general system of two linear equations in the unknowns x and y:

$$a_1x + b_1y = c_1$$

$$a_2x + b_2y = c_2$$

The graphs of these equations are lines; call them l_1 and l_2. Since a point (x, y) lies on a line if and only if the numbers x and y satisfy the equation of the line, the solutions of the

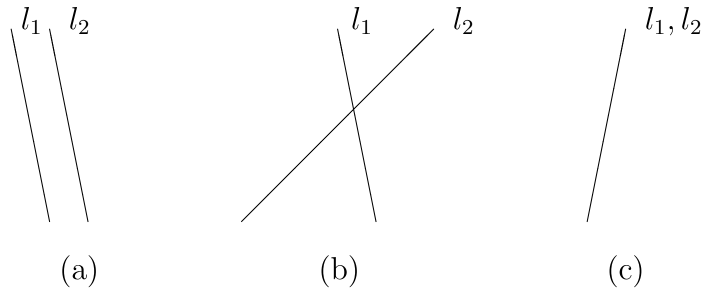


Figure 2.1: (a) no solution, (b) one solution, (c) infinitely many solutions

system of equations will correspond to points of intersection of l_1 and l_2. There are three possibilities:

The three possiblities illustrated in Figure 2.1 are as follows:

(a) l_1 and l_2 are parallel, in which case there is no intersection, and consequently no solution to the system.

(b) l_1 and l_2 intersect at only one point, in which case the system has exactly one solution.

(c) l_1 and l_2 coincide, in which case there are infinitely many points of intersection, and consequently infinitely many solutions to the system.

Although we have considered only two equations with two unknowns here, we will show later that this same result holds for arbitrary systems; that is, every system of linear equations has either no solutions, exactly one solution, or infinitely many solutions.

2.2 GAUSSIAN ELIMINATION

In this section we give a systematic procedure for solving systems of linear equations; it is based on the idea of reducing the augmented matrix to a form that is simple enough so that the system of equations can be solved by inspection.

Remark. It is not difficult to see that a matrix in row-echelon form must have zeros below each leading 1. In contrast a matrix in reduced row-echelon form must have zeros above and below each leading 1.

As a direct result of Figure 2.1 on page 26 we have the following important theorem.

Theorem 2.2.1. *A homogenous system of linear equations with more unknowns than equations always has infinitely many solutions*

The definition of matrix multiplication requires that the number of columns of the first factor A be the same as the number of rows of the second factor B in order to form the product AB. If this condition is not satisfied, the product is undefined. A convenient way to determine whether a product of two matrices is defined is to write down the size of the first factor and, to the right of it, write down the size of the second factor. If, as in Figure 2.2, the inside numbers are the same, then the product is defined. The outside numbers then give the size of the product.

Although the commutative law for multiplication is not valid in matrix arithmetic, many familiar laws of arithmetic are valid for matrices. Some of the most important ones and their names are summarized in the following proposition.

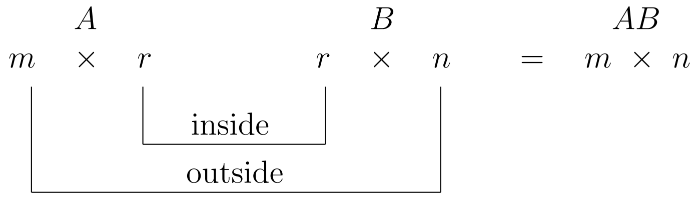


Figure 2.2: Inside and outside numbers of a matrix multiplication problem of $A \times B = AB$, showing how the inside dimensions are dropped and the dimensions of the product are the outside dimenions.

Proposition 2.2.2. *Assuming that the sizes of the matrices are such that the indicated operations can be performed, the following rules of matrix arithmetic are valid.*

(a) $A + B = B + A$ *(Commutative law for addition)*

(b) $A + (B + C) = (A + B) + C$ *(Associative law for addition)*

(c) $A(BC) = (AB)C$ *(Associative law for multiplication)*

2.3 FURTHER RESULTS ON SYSTEMS OF EQUATIONS

In this section we shall establish more results about systems of linear equations and invertibility of matrices. Our work will lead to a method for solving n equations in n unknowns that is more efficient than Gaussian elimination for certain kinds of problems.

2.3.1 Some Important Theorems

Theorem 2.3.1. *If A is an invertible $n \times n$ matrix, then for each $n \times 1$ matrix B, the system of equations $AX = B$ has exactly one solution, namely, $X = A^{-1}B$.*

Proof. Since $A(A^{-1}B) = B$, $X = A^{-1}B$ is a solution of $AX = B$. To show that this is the only solution, we will assume that X_0 is an arbitrary solution, and then show that X_0 must be the solution $A^{-1}B$.

If X_0 is any solution, then $AX_0 = B$. Multiplying both sides by A^{-1}, we obtain $X_0 = A^{-1}B$. □

Theorem 2.3.2. *Let A be a square matrix.*

(a) If B is a square matrix satisfying $BA = I$,then $B = A^{-1}$.

(b) If B is a square matrix satisfying $AB = I$, then $B = A^{-1}$.

In our later work the following fundamental problem will occur over and over again in various contexts.

Let A be fixed $m \times n$ matrix. Find all $m \times 1$ matrices B such that the system of equations $AX = B$ is consistent.

If A is an invertible matrix, Theorem 2.3.2 completely solves this problem by asserting that for every $m \times n$ matrix B, $AX = B$ has the unique solution $X = A^{-1}B$.

CHAPTER 3

EXAMPLES

Some examples of the definitions found in the file ps-defs.tex follow below.

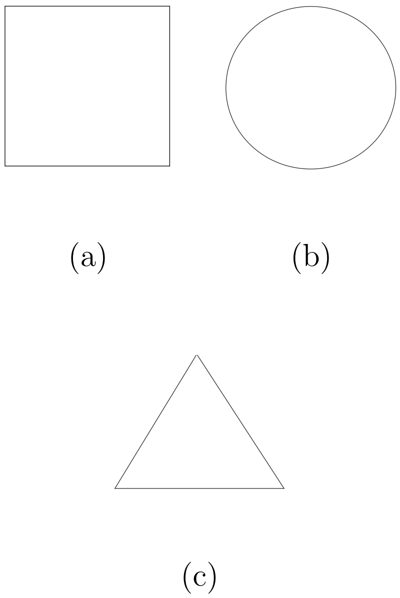
Here are examples of how you can use equation numbers with multiple line equations.

$$\begin{aligned}
(f+(g+h))(a) &= f(a)+(g+h)(a) \\
&= f(a)+(g(a)+h(a)) \qquad (3.1) \\
&= (f(a)+g(a))+h(a) \\
&= (f+g)(a)+h(a) \\
&= ((f+g)+h)(a) \qquad (3.2)
\end{aligned}$$

$$\begin{aligned}
(f+(g+h))(a) &= f(a)+(g+h)(a) \\
&= f(a)+(g(a)+h(a)) \\
&= (f(a)+g(a))+h(a) \qquad (3.3) \\
&= (f+g)(a)+h(a) \\
&= ((f+g)+h)(a)
\end{aligned}$$

$$\begin{aligned}
(f+(g+h))(a) &= f(a)+(g+h)(a) \\
&= f(a)+(g(a)+h(a)) \\
&= (f(a)+g(a))+h(a) \\
&= (f+g)(a)+h(a) \\
&= ((f+g)+h)(a)
\end{aligned}$$

Below is a figure which shows how to line up small figures on multiple lines. The .dvi version is immediately below. The .pdf version may be found underneath the complete figure and commented out. If you exchange the sections commented out, then you can compile a .pdf file.

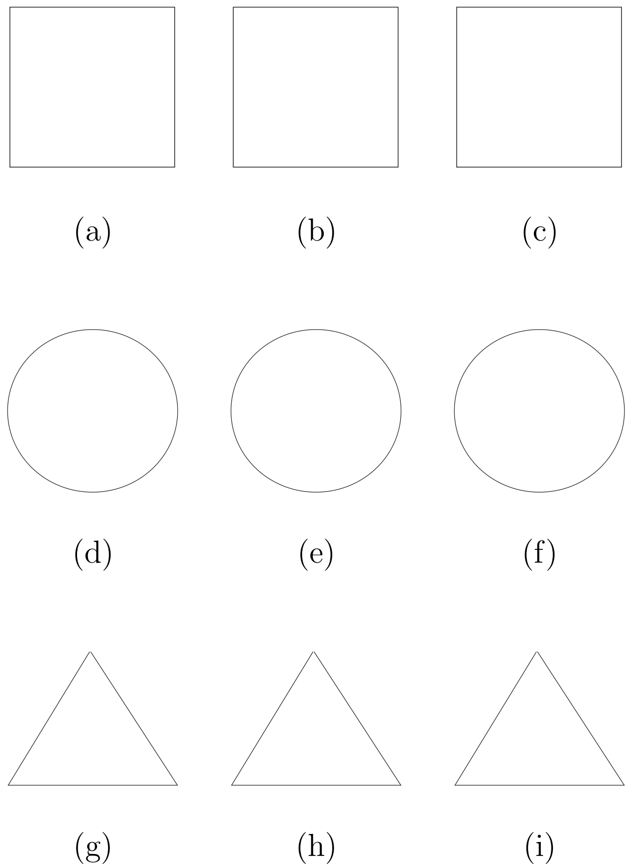


(a) (b)

(c)

Figure 3.1: Two rows of graphics: (a) Square (b) Circle (c) Rectangle

Three figures across the page requires fairly small figures to fit within the Graduate School margins.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.2: Three rows of graphics: (a)–(c) Squares. (d)–(f) Circles. (g)–(i) Ovals.

The verbatim environment can be useful when using data from a spreadsheet as is done below.

```
 X,TRUE_SUR,MSE SIM,MSE ZHAO,MSE JIAN,MSE PZHAO,MSE PJIAN
 0.0  0.7520  0.03864  0.01407  0.01407  0.01180  0.01223
 4.0  0.7273  0.04079  0.01675  0.01675  0.01479  0.01551
 8.0  0.7035  0.04203  0.01923  0.01923  0.01675  0.01817
12.0  0.6524  0.04581  0.02157  0.02135  0.01932  0.02043
16.0  0.6029  0.05146  0.02345  0.02266  0.02304  0.02320
20.0  0.5551  0.05343  0.02498  0.02393  0.02627  0.02509
24.0  0.5089  0.05449  0.02677  0.02453  0.02936  0.02641
28.0  0.4641  0.05706  0.02901  0.02442  0.03315  0.02722
32.0  0.4209  0.05719  0.02910  0.02341  0.03558  0.02776
36.0  0.3790  0.05656  0.02974  0.02229  0.03745  0.02667
40.0  0.3385  0.05518  0.02940  0.02119  0.03864  0.02618
44.0  0.2994  0.05344  0.02989  0.02054  0.03928  0.02531
48.0  0.2615  0.04950  0.02803  0.01906  0.03855  0.02414
52.0  0.2249  0.04582  0.02712  0.01812  0.03849  0.02229
56.0  0.1895  0.04101  0.02454  0.01578  0.03632  0.01918
60.0  0.1552  0.03564  0.02282  0.01315  0.03372  0.01629
64.0  0.1220  0.03216  0.02124  0.00997  0.03188  0.01391
68.0  0.0900  0.02420  0.01730  0.00688  0.02551  0.01070
72.0  0.0590  0.01592  0.01254  0.00363  0.01811  0.00622
76.0  0.0290  0.00865  0.00838  0.00110  0.00886  0.00368
```

Figure 3.3: Use of verbatim environment

On the following page is an example of how to rotate text that is too long to fit within the horizontal margins that are required.

$$A = \begin{pmatrix} -(\hat{\theta}_{D_1}(3;1) - \hat{\theta}_{D_1}(1;1)) & 0 & 0 & \cdots & 0 & 0 \\ (\hat{\theta}_{D_1}(3;1) - \hat{\theta}_{D_1}(1;1)) & -(\hat{\theta}_{D_2}(3;1) - \hat{\theta}_{D_2}(1;1)) & 0 & \cdots & \cdots & 0 \\ 0 & (\hat{\theta}_{D_2}(3;1) - \hat{\theta}_{D_2}(1;1)) & -(\hat{\theta}_{D_3}(3;1) - \hat{\theta}_{D_3}(1;1)) & \ddots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & (\hat{\theta}_{D_{n-1}}(3;1) - \hat{\theta}_{D_{n-1}}(1;1)) & -(\hat{\theta}_{D_n}(3;1) - \hat{\theta}_{D_n}(1;1)) \\ 0 & 0 & 0 & 0 & 0 & (\hat{\theta}_{D_n}(3;1) - \hat{\theta}_{D_n}(1;1)) \end{pmatrix},$$

$$\begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix},$$

Figure 3.4: Matrix Rotated 90 degrees.

REFERENCES

[1] D. R. Blum and J. G. Delgado-Frias. Schemes for eliminating transient-width clock overhead from set-tolerant memory-based systems. *IEEE Transactions on Nuclear Science*, 53(3):1564–1573, June 2006.

[2] T. Calin, M. Nicolaidis, and R. Velazco. Upset hardened memory design for submicron cmos technology. *IEEE Transactions on Nuclear Science*, 43(6):2874–2878, Dec 1996.

[3] N. Eftaxiopoulos, N. Axelos, and K. Pekmestzi. Donut: A double node upset tolerant latch. In *2015 IEEE Computer Society Annual Symposium on VLSI*, pages 509–514, July 2015.

[4] M. Fazeli, S. G. Miremadi, A. Ejlali, and A. Patooghy. Low energy single event upset/single event transient-tolerant latch for deep submicron technologies. *IET Computers Digital Techniques*, 3(3):289–303, May 2009.

[5] P. Hazucha, T. Karnik, S. Walstra, B. Bloechel, J. Tschanz, J. Maiz, K. Soumyanath, G. Dermer, S. Narendra, V. De, and S. Borkar. Measurements and analysis of ser tolerant latch in a 90 nm dual-vt cmos process. In *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003*, pages 617–620, Sept 2003.

[6] K. Katsarou and Y. Tsiatouhas. Soft error immune latch under seu related double-node charge collection. In *2015 IEEE 21st International On-Line Testing Symposium (IOLTS)*, pages 46–49, July 2015.

[7] K. Katsarou and Y. Tsiatouhas. Soft error interception latch: double node charge sharing seu tolerant design. *Electronics Letters*, 51(4):330–332, 2015.

[8] S. Lin, H. Yang, and R. Luo. High speed soft-error-tolerant latch and flip-flop design for multiple vdd circuit. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI '07)*, pages 273–278, March 2007.

[9] N. Miskov-Zivanov and D. Marculescu. Mars-c: modeling and reduction of soft errors in combinational circuits. In *2006 43rd ACM/IEEE Design Automation Conference*,

pages 767–772, July 2006.

[10] N. Miskov-Zivanov and D. Marculescu. Multiple transient faults in combinational and sequential circuits: A systematic approach. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(10):1614–1627, Oct 2010.

[11] P. C. Murley and G. R. Srinivasan. Soft-error monte carlo modeling program, semm. *IBM J. Res. Dev.*, 40(1):109–118, January 1996.

[12] M. Nicolaidis, R. Perez, and D. Alexandrescu. Low-cost highly-robust hardened cells using blocking feedback transistors. In *26th IEEE VLSI Test Symposium (vts 2008)*, pages 371–376, April 2008.

[13] M. Omana, G. Papasso, D. Rossi, and C. Metra. A model for transient fault propagation in combinatorial logic. In *On-Line Testing Symposium, 2003. IOLTS 2003. 9th IEEE*, pages 111–115, July 2003.

[14] M. Omana, D. Rossi, and C. Metra. High-performance robust latches. *IEEE Transactions on Computers*, 59(11):1455–1465, Nov 2010.

[15] Ramin Rajaei, Mahmoud Tabandeh, and Mahdi Fazeli. Single event multiple upset (semu) tolerant latch designs in presence of process and temperature variations. *Journal of Circuits, Systems and Computers*, 24(01):1550007, 2015.

[16] R. Rajaraman, J. S. Kim, N. Vijaykrishnan, Y. Xie, and M. J. Irwin. Seat-la: a soft error analysis tool for combinational logic. In *19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design (VLSID'06)*, pages 4 pp.–, Jan 2006.

[17] R. R. Rao, K. Chopra, D. T. Blaauw, and D. M. Sylvester. Computing the soft error rate of a combinational logic circuit using parameterized descriptors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(3):468–479, March 2007.

[18] F. Wang and Y. Xie. Soft error rate analysis for combinational logic using an accurate electrical masking model. *IEEE Transactions on Dependable and Secure Computing*,

8(1):137–146, Jan 2011.

[19] A. Yan, H. Liang, Z. Huang, and C. Jiang. High-performance, low-cost, and highly reliable radiation hardened latch design. *Electronics Letters*, 52(2):139–141, 2016.

[20] Bin Zhang, Wei-Shen Wang, and M. Orshansky. Faser: fast analysis of soft error susceptibility for cell-based designs. In *7th International Symposium on Quality Electronic Design (ISQED'06)*, pages 6 pp.–760, March 2006.

[21] M. Zhang, S. Mitra, T. M. Mak, N. Seifert, N. J. Wang, Q. Shi, K. S. Kim, N. R. Shanbhag, and S. J. Patel. Sequential element design with built-in soft error resilience. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(12):1368–1378, Dec 2006.

[22] M. Zhang and N. R. Shanbhag. Soft-error-rate-analysis (sera) methodology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(10):2140–2155, Oct 2006.

[23] Wei Zhao and Yu Cao. Predictive technology model for nano-cmos design exploration. *J. Emerg. Technol. Comput. Syst.*, 3(1), April 2007.

[24] J. F. Ziegler. Terrestrial cosmic rays. *IBM Journal of Research and Development*, 40(1):19–39, Jan 1996.

APPENDICES

APPENDIX I

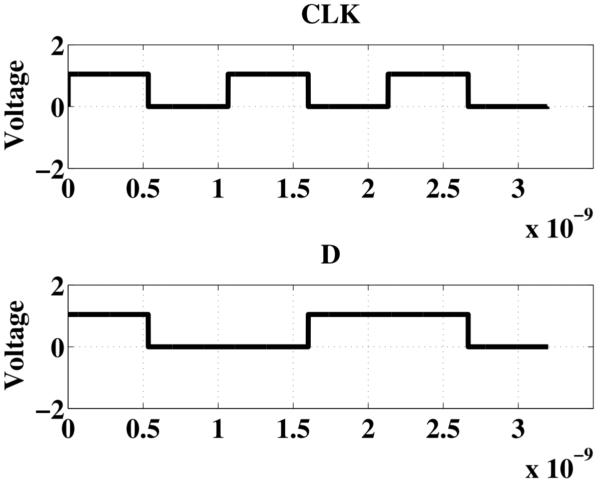


Figure 3.5: Waveforms for CLK and D.

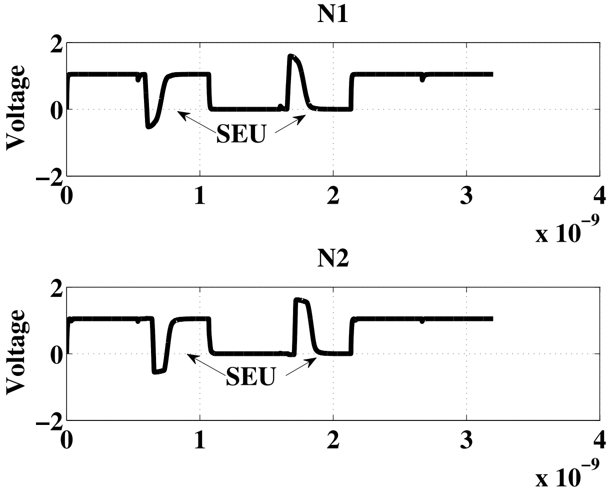


Figure 3.6: Node pair n1 and n2 upset and recovery.

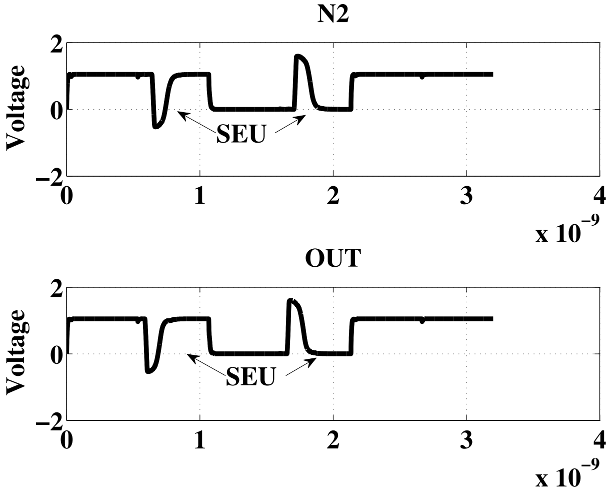


Figure 3.7: Node pair n2 and out upset and recovery.

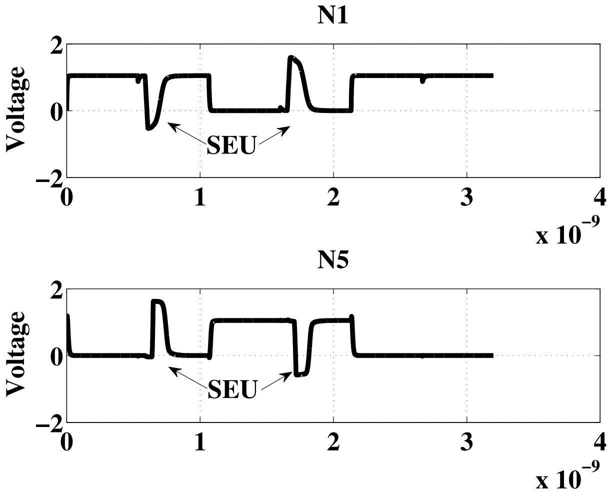


Figure 3.8: Node pair n1 and n5 upset and recovery.

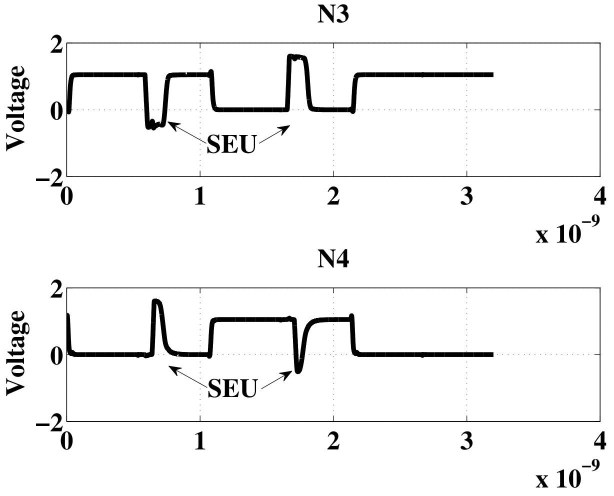


Figure 3.9: Node pair n3 and n4 upset and recovery.

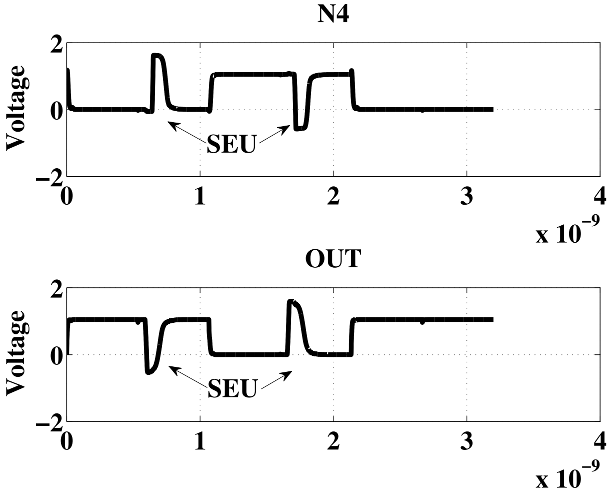


Figure 3.10: Node pair n4 and out upset and recovery.

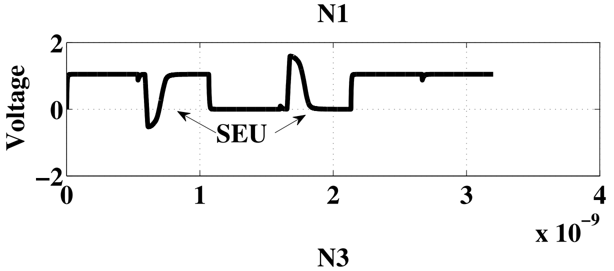


Figure 3.11: Node pair n1 and n3 upset and recovery.

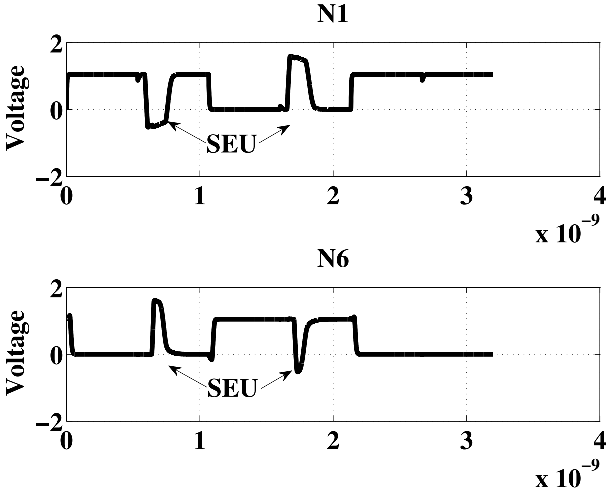


Figure 3.12: Node pair n1 and n6 upset and recovery.

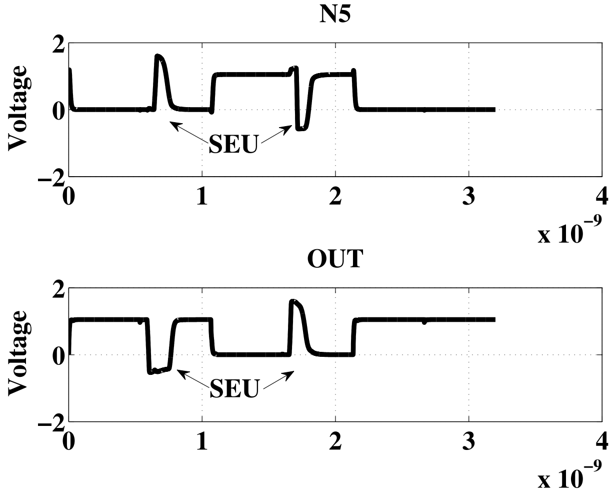


Figure 3.13: Node pair n5 and out upset and recovery.

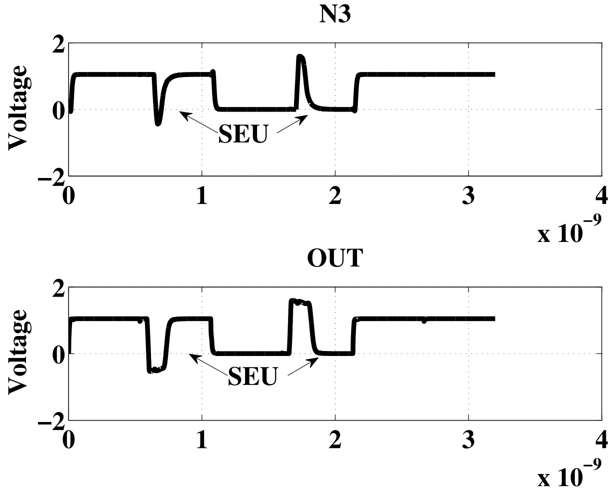


Figure 3.14: Node pair n3 and out upset and recovery.

APPENDIX II

VITA

Graduate School
Southern Illinois University

Adam Watkins Date of Birth: January 27, 1988

4158 Sycamore Unit A, Los Alamos, New Mexico, 87544

Email: acwatkins88@gmail.com

Southern Illinois University at Carbondale
Master's of Science, Electrical and Computer Engineering, May 2012

Southern Illinois University at Carbondale
Bachelor of Science, Electrical and Computer Engineering, May 2010

Special Honors and Awards: Best Student Paper Award, DFT 2016

Research Paper Title:
Analysis and Mitigation of Multiple Radiation Induced Errors in Modern Circuits

Major Professor: Dr. S. Tragoudas

Publications:

Anton, H., *Elementary Linear Algebra*, John Wiley & Sons, New York, 1977.

Huang, X. and Krantz, S.G., *On a problem of Moser*, Duke Math. J. **78** (1995), 213–228.