Radiation Hardened Latch Designs for Double and Triple Node Upsets

Adam Watkins, *Student Member, IEEE,* Spyros Tragoudas, *Senior Member, IEEE*

Abstract—As the process feature size continues to scale down, the susceptibility of logic circuits to radiation induced error has increased. This trend has led to the increase in sensitivity of circuits to multi-node upsets. Previously, work has been done to harden latches against single event upsets (SEU). Currently, there has been a concerted effort to design latches that are tolerant to double node upsets (DNU) and triple node upsets (TNU). In this paper, we first propose a novel DNU tolerant latch design. The latch is designed specifically to provide additional reliability when clock gating is used. Through experimentation, it is shown that the DNU tolerant latch is 11.3% more power efficient than latch designs suited for clock gating. In addition to the DNU tolerant design, we propose the first TNU tolerant latch. The TNU tolerant latch is shown to provide superior soft error resiliency while incurring a 40% overhead compared to DNU tolerant designs.

Index Terms—Soft Errors, Radiation Hardened Latch, Transient Pulses, Single Event Upset, Double Node Upset, Triple Node Upset.

1 Introduction

As the transistor feature size continuously scales down to improve performance, modern circuitry continues to become more susceptible to radiation induced errors commonly referred to as a soft error. Terrestrial soft errors can manifest from either neutron particles originating from cosmic rays or alpha particles from packaging. In space, a soft error mostly comes from protons, electrons, and heavy ions [1], [2]. A soft error of any source 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 propagated to a circuit output and captured by a flip-flop potentially causing an error. Furthermore, the error may occur directly on an internal latch of a flip-flop causing immediate data corruption. Due to this possibility, there is a need for new error tolerant latch designs.

There has been extensive research in the field of hardening latches against single even upsets (SEU). There have also been efforts in the development of hardened flip-flops [3]. This paper focuses on the latch since it can form the basis for more advanced flip-flop designs and does not directly suffer from the challenges of high setup and hold times. The most straight forward hardening latch design is the use of triple modular redundancy (TMR). This design consists of 3 standard latches connected to a 3-input majority voting circuit. While this design is robust 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 design is the DICE cell proposed in [4]. While this design works well for a single node, it is not capable of handling multi-node errors.

While the DICE latch is efficient in area, it suffers from high delay. For this reason there has been a multitude of alternative SEU tolerant devices that have been proposed. The SEU tolerant designs follow one of two approaches to hardening: sizing transistors such that the critical charge exceeds the maximum injected charge for the intended environment and by designing circuits that functionally tolerate the error. For the former designs, such as [5], they are typically performance and area efficient. The drawback with these types of designs is that they require accurate estimates for the maximum injected charge. If the maximum charge is found to be too high, a designer using this type of latch would have to choose between performance and reliability.

The latter type of latch, such as [6], [7], [8], [9], [10], [11], have the advantage of recovering from a SEU regardless of the injected charge due to the logical functions of the latch forcing recovery of affected node. In cases where the maximum injected charge is not excessively high, these latches have higher performance and area overheads compared to the previous type. However, these type of latches are preferable in many cases since the maximum charge may be unknown or very high.

In modern processes, the transistor size is small enough that a high energy radiation particle may simultaneously strike multiple transistors. Cases where this type of strike may occur are commonly referred to as a single event multiple upset (SEMU). In addition to the SEMU case, high flux environments may allow for the manifestation of a multiple event multiple upset (MEMU). In this case, multiple radiation particles strike internal transistors simultaneously. When either a SEMU or MEMU occur in a latch, they may upset multiple nodes. If two nodes are upset in the latch, this is referred to as a double node upset (DNU). If three nodes are upset, this is called a triple node upset (TNU).

- *Adam Watkins is with Los Alamos National Laboratory, Los Alamos, NM 87545 and Southern Illinois University Carbondale, Carbondale, IL 62901 E-mail: acwatkins88@lanl.gov*
- *Spyros Tragoudas is with Southern Illinois University Carbondale, Carbondale, IL 62901 E-mail: spyros@siu.edu*

Manuscript received March 1, 2017;

The DNU is currently of great concern as the feature size has allowed for a sharp increase in the occurrence of DNUs. Section 2 provides an overview of all existing DNU latches.

To save power, many modern circuit designs employ a technique commonly referred to as clock gating to further reduce the power consumption. Clock gating consists of shutting off the clock to a stable value or ”gating” the clock. If clock gating is used in a latch, it may need to hold the stored value for many clock cycles. If an error tolerant latch is struck by a radiation error while gated, it could lead to a loss of data. This may occur if the latch has high-impedance states after an error since the high-impedance nodes may slowly discharge causing a loss in data. To remedy this issue, researchers have proposed the addition of output circuitry to hold the data. However, as shown in Section 5, the additional circuitry adds a large overhead to the delay and power consumption.

To solve this problem, we propose, as first presented in [12], the HRDNUT (Highly Robust Double Node Upset Tolerant) latch which is an efficient DNU tolerant design that is capable of recovering all nodes after an error occurs. The recovery feature provides a distinct advantage over previous designs in cases where clock gating is used since it removes the need for additional circuitry since no nodes are held to a high-impedance state after an error. Designs that are DNU tolerant and exhibit this behavior are referred to as DNU-robust. Any design that is DNU tolerant and does has high-impedance states is referred to as DNU-non-robust. The proposed design is throughly compared to existing designs and is found to be more efficient than the existing DNU-robust design [13] in power, delay and area. The design is also compared to all existing DNU tolerant latches and the most common SEU tolerant latches.

In addition to the HRDNUT, we also propose the TNU-Latch which is a TNU tolerant latch that is based on the HRDNUT. While this latch is non-robust, it provides a simple and efficient solution suitable for high reliability applications with high radioactivity. To the authors’ knowledge, the TNU latch is the first of its kind. While many current research efforts focus on the development of SEU and DNU tolerant designs, it can be inferred that highly radioactive environments and environments with high energy particles will be vulnerable to TNUs. This provides the motivation for the design of a TNU tolerant latch.

The paper is organized as follows: Section 2 provides a discussion on existing DNU tolerant latches, Section 3 discusses the HRDNUT, Section 4 gives the TNU-latch, Section 5 contains a comparison of the proposed latches to many existing designs, and Section 6 concludes the paper.

2 Existing Multi-Error Tolerant Designs

In this section, we discuss the existing DNU tolerant designs and give a background for the TNU latch. First we will discuss the DNU tolerant designs. The first proposed design, designated as the DNCS (Double Node Charge Sharing) latch is presented in [14] and Fig. 3. It contains two DICE latches connected to a 2-input C-element. The DICE latch consists of 4 one-input c-elements connected in series. Two of the nodes are connected to a pass-gate which allows data to be loaded. Additionally, an example of the C-element is given in Fig. 1. The idea behind this design that it is impossible for a DNU to flip both DICE latches since they are SEU tolerant. More specifically, since the DICE latch is SEU tolerant, it requires a DNU to upset the data on a single cell. Even if a single cell is upset, the output C-element will hold the correct state. The output C-element only changes value when the inputs are unanimously the same value. Since both latches cannot be upset, the DNCS will tolerate a DNU. While this design is DNU tolerant, it is not DNU-robust since it may move to a high-impedance state after an error.

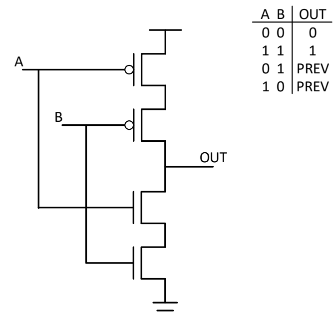

Fig. 1: Muller C-element

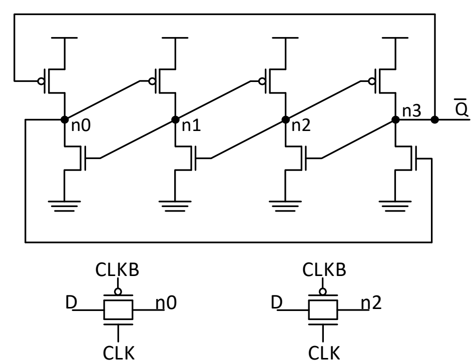

Fig. 2: DICE Latch.

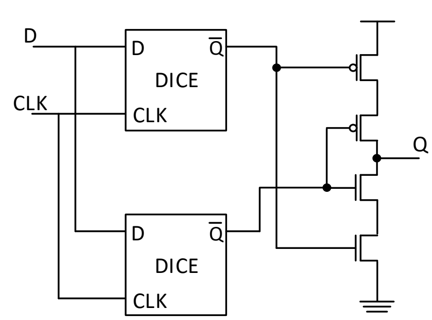

Fig. 3: The DNCS latch. A diagram of the DICE latch is given in Fig. 2.

Another design, named the interception latch and proposed in [15], improves on the DNCS by providing lower power consumption, delay and area. The latch functions using six 2-input C-elements connected in series. Every other node in the latch is fed to an output 3-input C-element. In this design, a DNU can only flip at most two nodes. Since the output is voted on by a C-element, it will not change value. Like the DNCS latch, a DNU will force the latch into a high-impedance state which implies the latch is DNU non-robust.

The most recent and efficient DNU tolerant latch is the HSMUF which is proposed in [16] and illustrated in Fig. 4. This latch uses the TP-DICE structure found in [17]. It is an extended DICE cell with a total of 6 nodes. In the TP-DICE cell, every other node is connected to the input of a 3-input Muller C-element. When a DNU occurs in the worst case, two nodes are set to erroneous value, two nodes are set to high-impedance and two nodes hold the correct value. Since a Muller C-element is placed on the output, the high-impedance and error-free nodes hold the output to the correct value. However, a drawback with this design is that it relies on high impedance states for reliability thus the design is non-robust. A common way to mitigate this issue is to place a weak-keeper at the output of the latch as in Fig. 4. While this design does ensure the output is held, the C-element must be sized such that the driving strength exceeds the that of the keeper. According to our simulations found in Section 5, the addition of the keeper substantially increases the delay, area and power consumption.

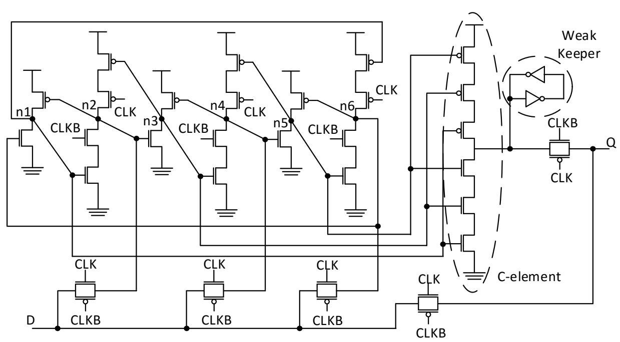

Fig. 4: HSMUF latch [16] with a weak keeper on the output.

As stated previously, latches that are capable of recovering all nodes after an error a called robust. This unique feature is desirable since it leads to more efficient latch designs that can recover from an error when the latch is clock gated. The most efficient existing DNU-robust design is the DONUT (Double Node Upset Tolerant) latch proposed in [13] as shown if Fig. 5. This latch is based on the combination of four DICE latches creating twelve nodes. As in Fig. 5, each node is connected to multiple cross coupled elements. In the diagram, a cross-coupled element is denoted by a box with an arrow. The arrow gives the direction of the element from input to out. The pass gates are given below the design where *D* represents the input to the latch and the given node number represents the node at which the pass-gate is connected.

Since the design is based on the DICE latch, it is able to exploit the recovery feature of the latch. One issue that was discovered during the testing of this latch is that it suffered from excessively high power and delay. It was found that the root of the problem was due to data contention on the data loading lines. To solve this issue, the latch was modified such that the data loading nodes were set to high impedance during the transparent mode. As shown in Section 5, this modification saved a large amount of power. This design is referred to as the DONUT-M and is given in Fig. 6.

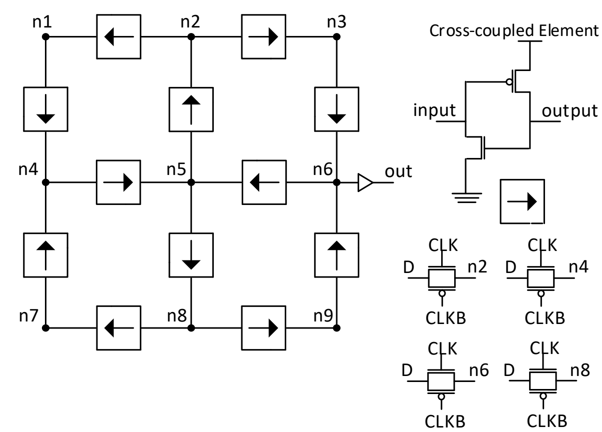

Fig. 5: DONUT latch as proposed in [13].

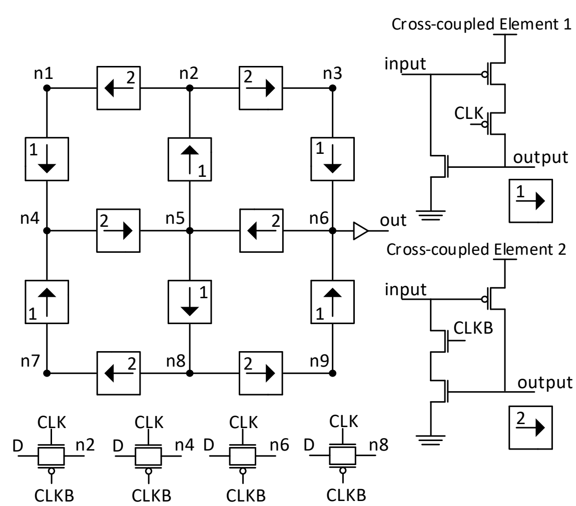

Fig. 6: Modified low-power DONUT latch.

In addition to DNU tolerant latches, we also investigate the TNU latch. The authors in [18] propose a latch design that shows limited TNU tolerance. Their design uses eight nodes and eight 2-input C-elements. There are four input signals which are each connected to the output of a C-element to load the data during the transparent mode. In the hold mode, the latch is fully tolerant to a DNU since at least one of the erroneous nodes will be driven by a C-element with error-free inputs. However, when a TNU occurs, the latch is only tolerant if all three errors are on adjacent nodes.

3 Discussion of HRDNUT Latch

This section discusses the HRDNUT latch. The HRDNUT latch is based on a basic storage block given in Fig. 7 which contains a 3-input C-element connected to an inverter. This design is derived from a standard keeper element. In addition, the C-element contains two transistors driven by CLK and CLKB. The purpose of the transistors are to set the output to high impedance during the transparent mode. As demonstrated by the DONUT-M, the addition of these transistors drastically reduces the power and delay at a cost of area. Once in the transparent mode, the output is loaded using a pass-gate which allows D to be set directly.

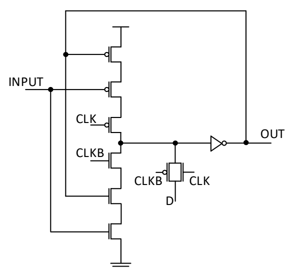

Fig. 7: Basic data storage loop block.

The basic block is then inter-connected to form the block based latch in Fig. 8. To ensure that the nodes recover after an error, each C-element in the data loop is driven by the other two nodes. This type of configuration ensures that an error on a C-element output is not held. The latch in Fig. 8 is SEU tolerant and DNU tolerant when one error strikes the output C-element. While this latch is not DNU tolerant, it forms the basis for the DNU robust design. To demonstrate the functionality of the latch, we will first describe its operation during normal operation.

In normal operation, the latch data is loaded during the transparent mode. In this mode the input clock *CLK* is high and the inverted clock, *CLKB*, is low. At this stage, nodes *n1*, *n2* and *n3* are set to high impedance and each pass gate is activated to allow the value at D to be loaded to the corresponding node. Once loaded, the data will propagate to the inputs of the output C-element. During the hold mode, *CLK* is set to a low value and *CLKB* is set high. The held data is reinforced by C-elements *C1*, *C2* and *C3*. The output is then set since the inputs of the output C-element are unanimously held to a single value.

In the case of an SEU in an internal node on the block based latch, the error will propagate to the other C-elements. However, since each C-element has at least one unaffected node, the data on the node will fully recover allowing full recovery of the latch state. In the case of a DNU on the internal nodes, the inputs of the unaffected C-element will be flipped due to the errors. This will ultimately flip the output leading to an error on the output. While this latch is not DNU tolerant, as stated before, it forms the basis for the HRDNUT.

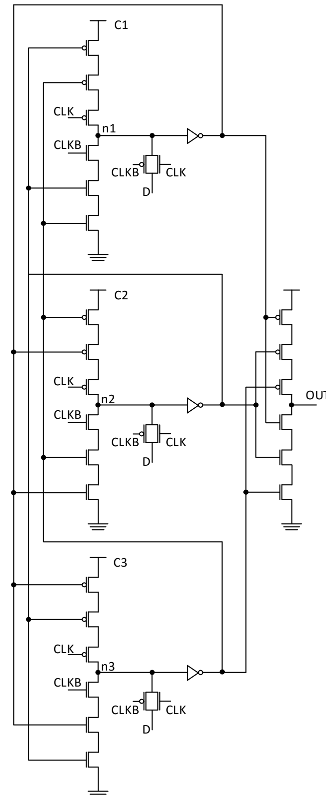

Fig. 8: Schematic of the block-based latch.

The HRDNUT latch is realized by modifying the block latch such that no C-element drives itself. More specifically, we first add additional resiliency to the latch by transforming the inverter on node *n2* to a two-input C-element. The second input on this element is driven by the output. This allows for an error on *n2* to be blocked by C-element *C4*. Next, to add resiliency to nodes *n1* and *n3*, elements *C5* and *C6* are added. In addition to DNU tolerance, the C-elements allow for robustness of the latch by ensuring that *C1*, *C2* and *C3* do not drive themselves. Note that on *C5* and *C6*, one NMOS and one PMOS transistor is driven by *OUT* and *n2*. This ensures that the latch will recover all nodes since the node *n5* drives the PMOS on *C7* while the output of *C7* drives the PMOS on *C5*. The idea behind this is that *n5* only affects C-element *C7* if *n5* is a low value. However, an error on *OUT* does not prevent recovery since *C7* is only affected if *n5* is 0. *C6* operates similarly but node *OUT* drives a NMOS. A schematic of the HRDNUT is given in Fig. 10.

Now that we have explained the design process, we will evaluate the HRDNUT latch during normal operation as in [12]. 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 output 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. 9 provides the waveforms of the CLK, D and OUT nodes for both the transparent and hold modes of operation.

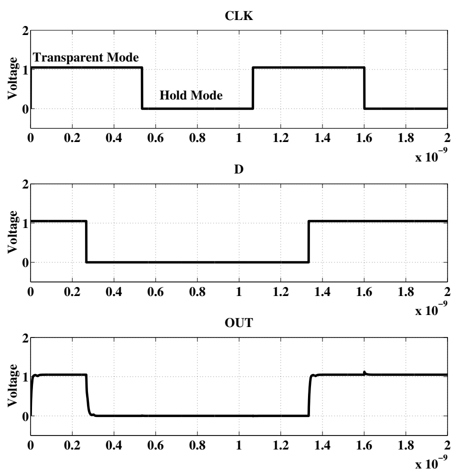

Fig. 9: Waveforms of the HRDNUT latch during normal operation.

In the case of an SEU, the HRDNUT retains the SEU resiliency of the block based latch and adds 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, we will evaluate the latch 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 our analysis, we categorize the possible DNU strike combinations into 9 distinct cases based on their effect in the HRDNUT latch. The categories are discussed in detail below.

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*).

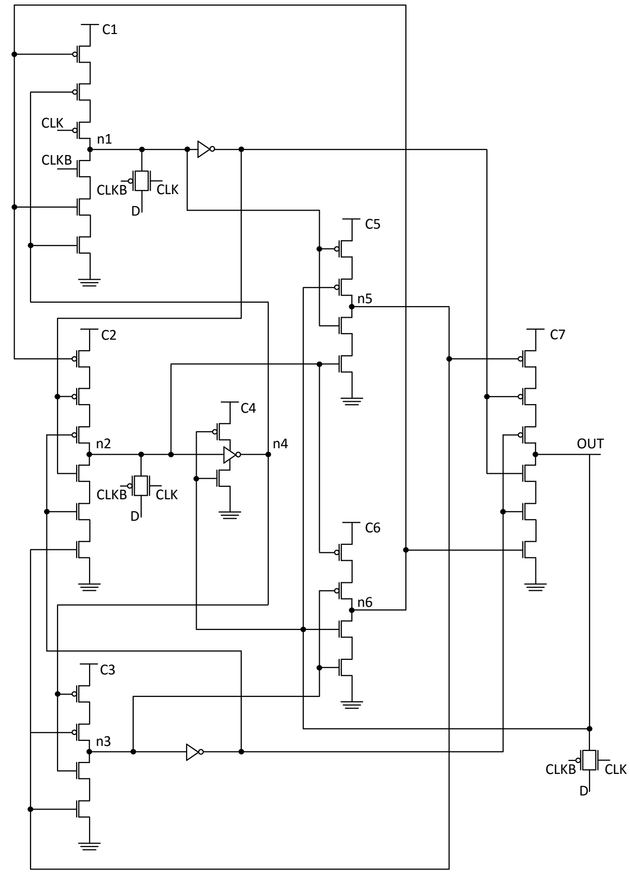

Fig. 10: Schematic of the HRDNUT latch.

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*. 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.

To evaluate the design, pulses were injected using the equation given in [19]. The equation is given below with τ as the technology dependent constant, Q_o as the injection charge and t as the variable for time. In the equation τ was set to 32×10^{-12} and Q_o was set to 5fC. The charge was selected such that the pulses were sufficiently large to cause an upset on the respective node. Other charges were not tested since the number of pulses, not the magnitude, contribute to the reliability of the latch. The results of this simulation for each distinct case is given in Figs. 11-20, respectively.

$$I(t) = \frac{2Q_o}{\tau\sqrt{\pi}}\sqrt{\frac{t}{\tau}}e^{\frac{-t}{\tau}} \tag{1}$$

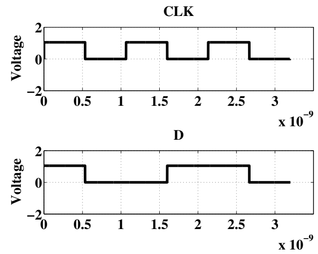

Fig. 11: Waveforms for CLK and D.

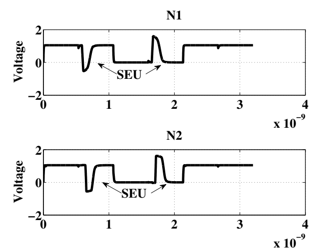

Fig. 12: Node pair n1 and n2 upset and recovery.

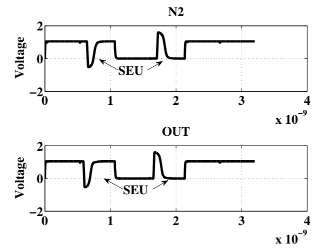

Fig. 13: Node pair n2 and out upset and recovery.

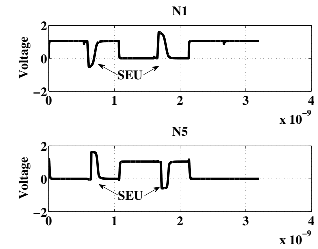

Fig. 14: Node pair n1 and n5 upset and recovery.

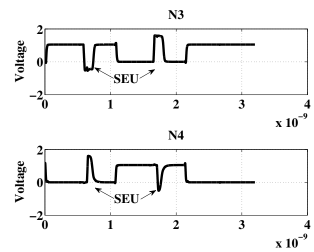

Fig. 15: Node pair n3 and n4 upset and recovery.

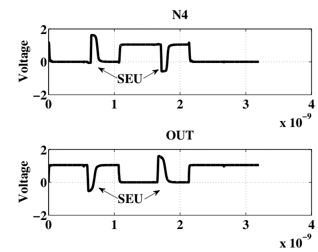

Fig. 16: Node pair n4 and out upset and recovery.

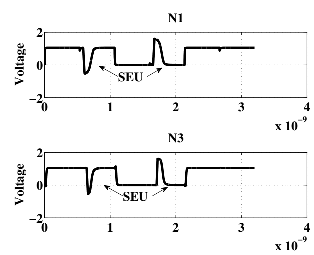

Fig. 17: Node pair n1 and n3 upset and recovery.

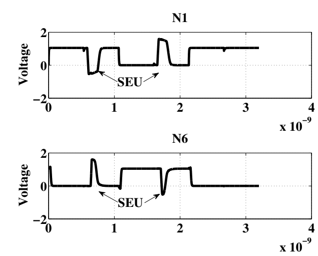

Fig. 18: Node pair n1 and n6 upset and recovery.

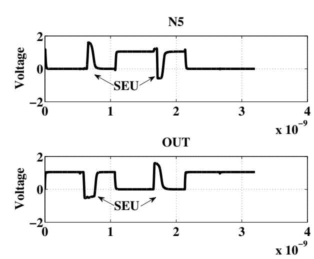

Fig. 19: Node pair n5 and out upset and recovery.

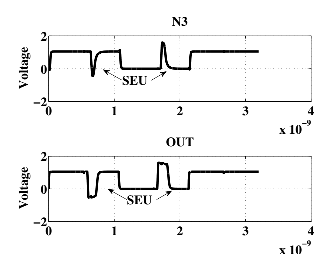

Fig. 20: Node pair n3 and out upset and recovery.

4 Proposed TNU Tolerant Latch

In this section we discuss the implementation of the non-robust triple node upset (TNU) tolerant latch named TNU-latch. We have investigated the development of a robust TNU latch but it was exceptionally difficult to verify its correctness. For example, a latch with two more nodes compared to the HRDNUT resulted into nine total nodes and requires the verification of 84 unique cases. Instead, we focused on the development of a simple and efficient non-robust design that has 82 transistors. This design has been verified for all possible TNU cases and is shown to be fully tolerant.

A schematic of the TNU-latch is given in Fig. 21. The latch consists of a base block latch as in the HRDNUT but with 5 storage blocks. Each storage block has a C-element with 4 inputs with each input connected to the other nodes. In addition four of the C-elements have two transistors connected to *CLK* and *CLKB* to ensure that the output node is set to high impedance during the transparent mode. Similar to the HRDNUT, this reduces the power consumption and the delay for a relatively small increase in area. To vote on the output, the nodes in the block latch are connected to two 3-input C-elements which are denoted as *C6* AND *C7* in 21. These C-elements drive a 2-input C-element labeled as *C8*. A schematic of the TNU-latch is given in Fig. 21.

The basis behind the latch design is that the C-elements in the block latch cannot be driven to an incorrect value due to a TNU. To ensure this, each C-element has four inputs. If the latch was designed as in the block based latch for the HRDNUT, the output C-element would had five inputs. However, after experimentation with this design it was found that an error on the output element would lead to an unrecoverable error. To solve this issue, the output C-element was split into two 3-input elements which drive a 2-input element. This removed the error since a TNU can, in the worst case, only flip C-element *C6* or *C7* leaving one C-element unaffected thus holding the data. This also allows for the latch to tolerate a TNU with an error on the output since only two errors will affect the internal nodes. None of the C-elements can be flipped due to an internal DNU, thus allowing for the output to recover.

First, we will evaluate the TNU-latch during the transparent mode. In this mode, the data is loaded to nodes *n1*, *n2*, *n3* and *n4* (see Fig. 21). This is done when the clock is at a high value which sets the output node to high impedance and turns on the loading pass gates. Once the four nodes are loaded, all the inputs on C-element *C5* are set such that *n5* is set to the loaded value. Since nodes *n1-n5* are all loaded, C-elements *C6* and *C7* are set thus driving *C8*.

We will now evaluate the latch for tolerance against soft errors. In the case of a SEU, the latch is tolerant since an SEU cannot change the state of a C-element. Additionally, the latch is DNU tolerant for similar reasons. When a TNU occurs, there are 56 total strike cases. Due to the simple design of the latch, we condense all cases into 6 distinct cases which are given in the following list. To verify the cases, waveforms were generated using equation 1 and the same simulation parameters as in Section 3 to model the pulse shape for each individual case. The waveforms are given in Figs. 22-27.

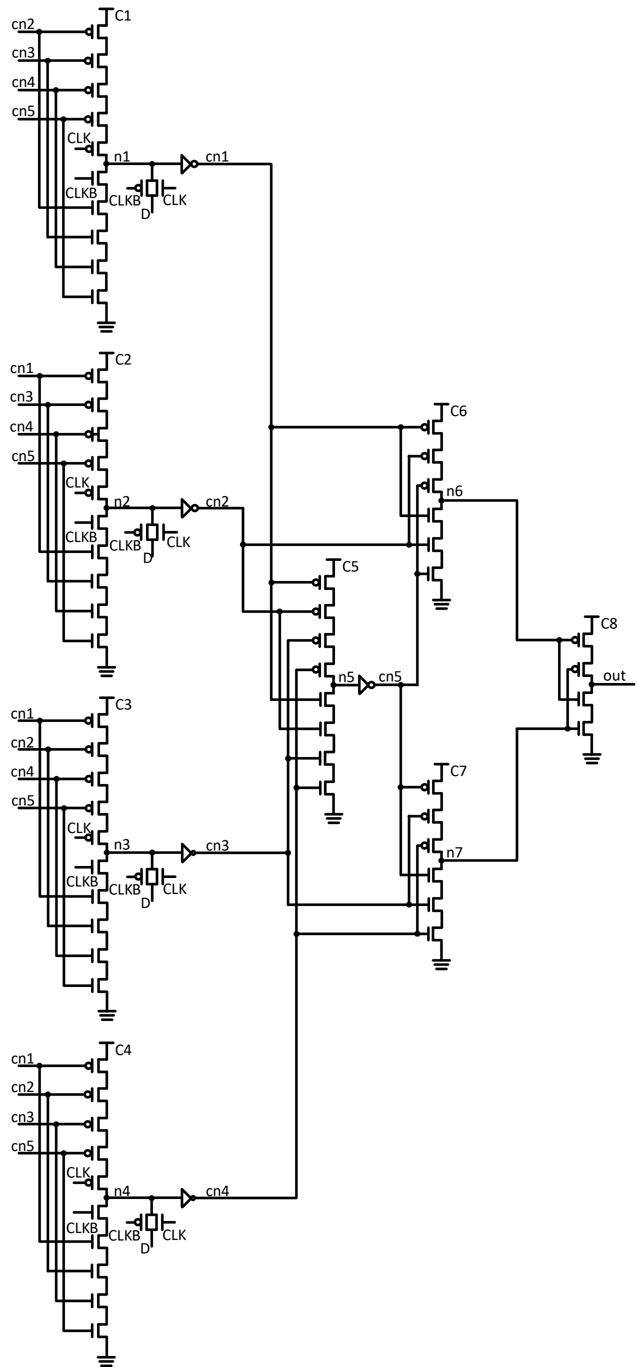

Fig. 21: Schematic of the TNU-Latch.

1) This case considers any three strikes on the set of nodes [*n1, n2, n3, n4, n5*]. To demonstrate this case, we consider strikes on nodes *n1*, *n2* and *n3*. While we only discuss this single case, this explanation applies to any combination of nodes that fall in this case. The errors will all propagate to C-element *C5* but will not cause a change on node *n5* since *n4* is not affected. Additionally, the errors will propagate to the inputs of *C1*, *C2*, *C3* and *C4*. However, since at least *n5* will be unaffected, the C-elements will hold their state. Next, we look at C-elements *C6* and *C7* and note that since *n5* does not change values, the C-elements hold the correct value on nodes *n6* and *n7*. Since these nodes have the correct value, node *out* does not change.
2) This case considers a strike on node *out* and any combination of two strikes on nodes within the set [*n1, n2, n3, n4, n5*]. We present the case where nodes *n1*, *n2* and *out* are struck by a TNU. The other instances of TNU for this case will evaluate similarly. The errors on the block latch can be treated as a DNU. Since no additional nodes in the block latch are flipped by the error, the C-elements *C6* and

C7 can only be affected by at most two errors. This implies that neither element will possibly flip due to an error. Since nodes *n6* and *n7* are error-free, *C8* will drive *out* back to the correct state.

3) This case consists of all TNU strikes where the strike affects a single node within the set [*n1, n2, n3, n4, n5*] and strikes on nodes *n6* and *n7*. For simplicity, we only consider the case for errors on *n1*, *n6* and *n7*. All other strike combinations that fall under this case will evaluate similarly. The error on *n1* will propagate to internal elements *n2, n3, n4* and *n5*. However, since only a single error is at the inputs of the C-elements they hold their correct value. Additionally, *C1* is driven back to the error-free value. This allows for nodes *n6* and *n7* to also be recovered.
4) This case applies to any TNU combination that has two errors from the set [*n1, n2, n3, n4, n5*] and a single error from *n6* or *n7*. As in the previous cases, any TNU falling within this strike set will evaluate similarly. We present this case by analyzing a TNU striking nodes *n1*, *n2* and *n6*. The errors on *n1* and *n2* propagate to C-elements *C1*, *C2*, *C3*, *C4* and *C5*. Since each C-element is driven by 4 nodes, no additional nodes flip value. This leads to two of the nodes on *C6* having two erroneous inputs. This ensures that the error on *n6* is not recovered. However, *n7* remains error-free thus allowing the output to be fully recovered.
5) This case applies to any TNU instance which has one error in the set of [*n1, n2, n3, n4, n5*], a single error on *n6* or *n7*, and an error on *out*. To present the analysis consider errors on nodes *n1*, *n6* and *out*. In this case the error on *n1* will be fully recovered as in case 3. Since the error is recovered, all input nodes to *C6* are error-free allowing for full recovery of the node. *n6* and *n7* are also error-free setting the output of *C8* to the correct value.
6) Lastly, assume errors on *n6, n7* and *out*. The TNU-latch will fully recover since all of the inputs of *C6* and *C7* are error-free driving nodes *n6* and *n7* to the correct value. The nodes then drive the output of *C8* to the previous error-free value.

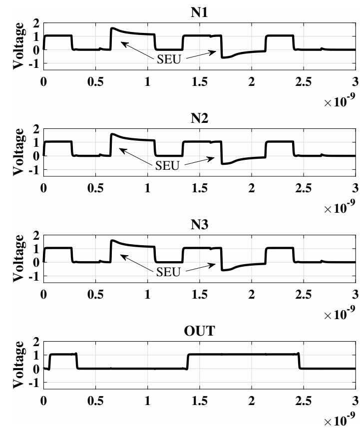

Fig. 22: Waveforms for case 1.

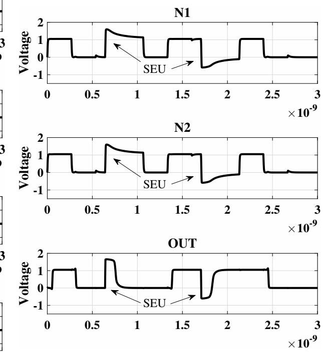

Fig. 23: Waveforms for case 2.

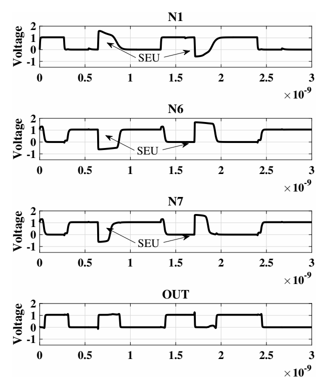

Fig. 24: Waveforms for case 3.

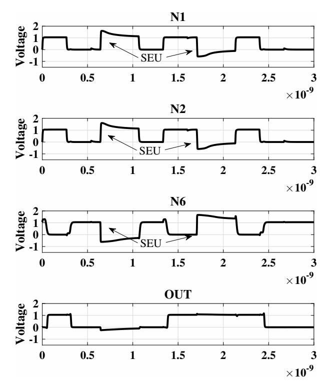

Fig. 25: Waveforms for case 4.

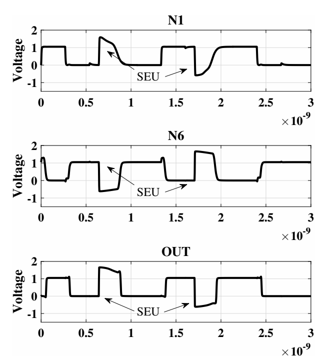

Fig. 26: Waveforms for case 5.

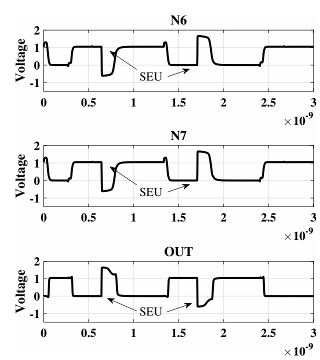

Fig. 27: Waveforms for case 6.

5 Comparative Study to Existing Designs

The proposed latch designs were implemented using the 1.05V 32nm PTM library [20] and simulated in HSPICE. All transistor widths for all designs were set to minimum size which is 80nm for PMOS and 40nm for NMOS. All designs were operated at 1 Ghz. We compared the HRDNUT and TNU-latch to existing SEU and DNU tolerant designs. We did not compare to other TNU tolerant designs since no other designs are known to exist. For the analysis, we compared to the following SEU tolerant latches: DICE [4], FERST [7] and HIPER [6]. Additionally, we compared to the following DNU tolerant designs: DNCS [14], Interception [15], HSMUF [16] and DONUT [13]. 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 320nm and the NMOS width was set to 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 the output to an erroneous value in the event of an error.

We measured the propagation delay, average power consumption, critical charge and area of all designs. We then categorized the designs based on the number of errors they can tolerate and if they are DNU-robust. The delay for each design was calculated based on the difference between the time that input *D* is at $0.5 * V_{DD}$ and the output at the same point (D->Q delay). The average power was calculated over

TABLE 1: SPICE Simulations of Existing Latches using the 1.05V 32nm PTM library

Latch	DNU Immune	DNU Robust	TNU Immune	Power (μW)	D->Q Delay (ps)	Area (UST)
DICE	No	No	No	1.332	8.145	16
FERST	No	No	No	3.178	31.648	60
HIPER	No	No	No	1.292	2.221	27
DNCS	Yes	No	No	4.948	22.486	61
[15]	Yes	No	No	5.606	79.168	89
HSMUF	Yes	No	No	1.871	1.0626	51
HSMUF (Keeper)	Yes	No	No	3.787	3.945	78
DONUT [13]	Yes	Yes	No	4.021	14.722	54
DONUT-M	Yes	Yes	No	2.760	8.421	72
HRDNUT (Proposed)	Yes	Yes	No	2.450	2.310	66
TNU-Latch	Yes	No	Yes	3.899	46.89	123

a 200 ns duration when the latch is error-free. All latches tested do not have a critical charge in a conventional sense. Specifically, it is the number of injected pulses rather than the magnitude that greatly contributes to the SEU, DNU or TNU tolerance of the design. We attempted to find the charge value for this to occur through HSPICE simulation of injected charges up to 1 pC and found that the latches did not flip. Waveforms showing the results of this simulation for the injection of a 1 pC DNU on the HRDNUT and a 1 pC TNU for the TNU-Latch are given in Figs. 28 and 29, respectively. For the calculation of the area, the unit size transistor (UST) metric as adopted in [14] was used. This metric quantifies the expected area based the total transistor area divided by the unit size. For this case, the unit size was set to be 40 nm. Table 1 gives the results of the simulation.

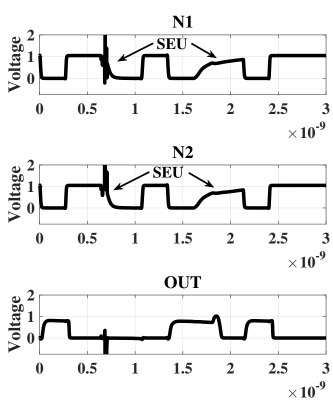

Fig. 28: Simulation of the HRDNUT for a 1 pC DNU on nodes N1 and N2.

According to Table 1, the DNU robust designs tested were the two DONUT latches and the HRDNUT. In comparison to the improved DONUT-M latch, the HRDNUT had similar robustness with 11.3% lower power consumption, 8.33% fewer transistors and a 72.5% lower propagation delay. Furthermore, when the HRDNUT was compared to the HSMUF with a keeper, it consumed substantially less power with a lower area and delay. Considering the compared latches, it can be observed that the HRDNUT is the best option for clock gating applications due to DNU robustness

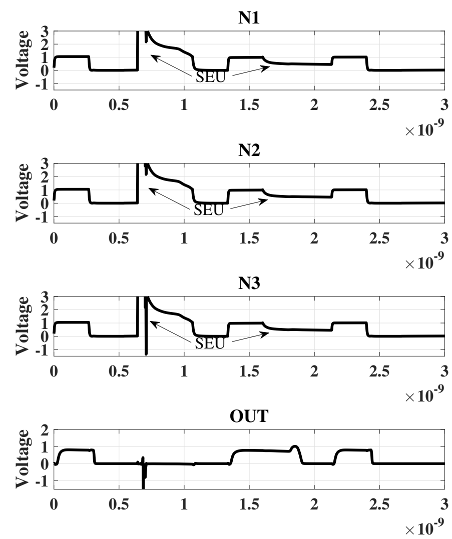

Fig. 29: Simulation of the TNU-Latch for a 1 pC TNU on nodes N1, N2 and N3.

and lower power, delay and area overheads.

In addition to the HRDNUT, the TNU-Latch was also compared to many existing designs. While the TNU-Latch does have the highest overhead, it is to be expected due to the additional circuitry that TNU tolerance requires. Compared to the HRDNUT, the TNU-Latch consumes approximately 40% more power while costing about 2X more area and 20X more delay. The increase in delay is manageable since it is still in the pico-seconds (ps) range which allows the latch to be driven at any commonly used frequency. Additionally, the TNU-Latch still has less delay than the DNU tolerant interception latch [15]. Lastly, compared to all other designs, the TNU-Latch is the only latch that provides full TNU resiliency.

6 Conclusion

In this paper, novel DNU and TNU tolerant latches were proposed. The proposed HRDNUT latch provided DNU robustness which allows for the design to be used in clock gating schemes. Existing designs that were used in clock gating typically relied on the addition of a weak keeper on the output of the latch. As shown in this paper, this circuitry greatly increases the power, area and delay. The only exception to the weak keeper was the DONUT latch which provides DNU robustness. It was shown in Section 5 that the HRDNUT is more efficient compared to the DONUT by providing 11.3% lower power consumption, 8.33% lower delay and 72.5% lower propagation delay.

Furthermore, the TNU-Latch was introduced which is the first fully TNU tolerant design. Compared to the HRDNUT, the TNU-Latch has a 2X area overhead and consumes 40% more power. Based off these findings, it is recommended to use the TNU-Latch in extreme radiation environments for modern process technologies. However, as the transistor feature size continues to shrink, the likelihood of a TNU may become a significant concern for terrestrial designs. In addition to terrestrial applications, there has been a concerted effort to send electronics using modern transistor sizes to space due to the fact that most space-grade hardware is on the order of 10-20 years out of date. Highly robust designs, such as the TNU-Latch, may help alleviate this issue by allowing small transistor processes to be used in current processor designs.

REFERENCES

[1] K. M. Zick and J. P. Hayes, "High-level vulnerability over space and time to insidious soft errors," in *2008 IEEE International High Level Design Validation and Test Workshop*, Nov 2008, pp. 161–168.

[2] J. R. Schwank, M. R. Shaneyfelt, and P. E. Dodd, "Radiation hardness assurance testing of microelectronic devices and integrated circuits: Radiation environments, physical mechanisms, and foundations for hardness assurance," *IEEE Transactions on Nuclear Science*, vol. 60, no. 3, pp. 2074–2100, June 2013.

[3] S. Shambhulingaiah, C. Lieb, and L. T. Clark, "Circuit simulation based validation of flip-flop robustness to multiple node charge collection," *IEEE Transactions on Nuclear Science*, vol. 62, no. 4, pp. 1577–1588, Aug 2015.

[4] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron cmos technology," *IEEE Transactions on Nuclear Science*, vol. 43, no. 6, pp. 2874–2878, Dec 1996.

[5] 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)*, April 2008, pp. 371–376.

[6] M. Omana, D. Rossi, and C. Metra, "High-performance robust latches," *IEEE Transactions on Computers*, vol. 59, no. 11, pp. 1455–1465, Nov 2010.

[7] 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*, vol. 3, no. 3, pp. 289–303, May 2009.

[8] 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*, Sept 2003, pp. 617–620.

[9] R. Rajaei, M. Tabandeh, and M. Fazeli, "Single event multiple upset (semu) tolerant latch designs in presence of process and temperature variations," *Journal of Circuits, Systems and Computers*, vol. 24, no. 01, p. 1550007, 2015.

[10] S. Lin, H. Yang, and R. Luo, "types o," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI '07)*, March 2007, pp. 273–278.

[11] 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*, vol. 14, no. 12, pp. 1368–1378, Dec 2006.

[12] A. Watkins and S. Tragouodas, "A highly robust double node upset tolerant latch," in *2016 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Sept 2016, pp. 15–20.

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

[14] K. Katsarou and Y. Tsiatouhas, "Double node charge sharing seu tolerant latch design," in *2014 IEEE 20th International On-Line Testing Symposium (IOLTS)*, July 2014, pp. 122–127.

[15] ——, "Soft error immune latch under seu related double-node charge collection," in *2015 IEEE 21st International On-Line Testing Symposium (IOLTS)*, July 2015, pp. 46–49.

[16] A. Yan, H. Liang, Z. Huang, and C. Jiang, "High-performance, low-cost, and highly reliable radiation hardened latch design," *Electronics Letters*, vol. 52, no. 2, pp. 139–141, 2016.

[17] 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*, vol. 53, no. 3, pp. 1564–1573, June 2006.

[18] ——, "Hardened by design techniques for implementing multiple-bit upset tolerant static memories," in *2007 IEEE International Symposium on Circuits and Systems*, May 2007, pp. 2786–2789.

[19] J. F. Ziegler, "Terrestrial cosmic rays," *IBM Journal of Research and Development*, vol. 40, no. 1, pp. 19–39, Jan 1996.

[20] W. Zhao and Y. Cao, "Predictive technology model for nano-cmos design exploration," *J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 1, Apr. 2007. [Online]. Available: http://doi.acm.org/10.1145/1229175.1229176

Adam Watkins received his B.S. degree in computer engineering and M.S. and Ph.D. degrees in electrical and computer engineering from Southern Illinois University Carbondale, Carbondale, IL in 2010, 2012 and 2016, respectively.

He is currently a Post-Doctoral Research Associate in the Intelligence and Space Research (ISR) division at Los Alamos National Laboratory, Los Alamos, NM. His current research interests are in the fields of radiation effects on electronics and charging on spacecraft.

Spyros Tragoudas received his Diploma in Computer Engineering from the University of Patras, Greece in 1986, and M.S. and PhD degrees in Computer Science from the University of Texas at Dallas in 1988 and 1991. He is Professor and Department Chair at the Electrical and Computer Engineering (ECE) Department, Southern Illinois University at Carbondale (SIUC), and the Director of the National Science Foundation (NSF) Industry University Cooperative Research Center (IUCRC) on Embedded Systems at the SIUC site. He has held prior appointments with the faculty of the ECE Department at the University of Arizona, and with the faculty of the Computer Science Dept. at SIUC.

His current research interests are in the areas of VLSI design/test automation and embedded systems. Dr. Tragoudas has published over two hundred and fifty papers in peer-reviewed journals and conference proceedings as well as book chapters in these areas. He has received three outstanding paper awards, and has directed many PhD dissertations and MS theses. His research has been funded by federal agencies and industry. He has served and current serving on the editorial board of several journals, the technical program committees of many conferences, was the program chair of DFTS09, and the general chair of DFTS10.