A Highly Robust Double Node Upset Tolerant Latch

Adam Watkins*
Los Alamos National Laboratories
Los Alamos, NM 87545
Email: acwatkins88@lanl.gov

Spyros Tragouodas
Southern Illinois Univeristy Carbondale
Carbondale, IL 62901
Email: spyros@engr.siu.edu

***Abstract*—Due to technology scaling, radiation induced errors which cause a double node upset (DNU) have become more common in data storage elements. All current designs either suffer from high area and performance overhead or are vulnerable to an error after a DNU thus making them unsuitable for clock gating. A novel latch design is proposed in which all internal and external nodes are capable of recovering the previous value after a single or double node upset. The proposed latch offers higher speed, lower power consumption and lower area requirements compared to all existing DNU tolerant latches capable of recovering all nodes.**

I. Introduction

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 paper, we focus 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 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 cell is the DICE cell proposed in [1]. The design in [1] 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 [2]–[8].

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 II.

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 vulnerable state in which the correct state 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 paper, 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 work, we propose the HRDNUT (Highly Robust Double Node Upset Tolerant) latch which is a robust latch design that recovers all nodes to the previous state after a DNU occurs while maintaining lower power and area overheads compared to existing DNU robust designs. The paper is

*Adam Watkins is also with Southern Illinois University Carbondale

This research has been supported in part by grants NSF IIP 1432026, and NSF IIP 1361847 from the NSF I/UCRC for Embedded Systems at SIUC. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation

organized as follows: Section II consists of a discussion on existing DNU tolerant designs. Section III provides the design of the HRDNUT. Section IV consists of a comparison of the HRDNUT to existing error tolerant latch designs. In Section V we conclude the paper.

II. Existing DNU Tolerant Designs

Currently, there are a few existing DNU tolerant designs. The first proposed design found in [9], 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 [10] propose an enhanced design compared to [9]. 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 [11] and is referred to as the HSMUF latch. Fig. 1 provides the design. The HSMUF uses the TP-DICE [12] 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, 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. 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 IV, 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.

To the author's knowledge, the existing most efficient DNU robust design capable of recovering all nodes after a DNU is the DONUT latch [13] in Fig 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 IV). The reason for the high power consumption is due to contention on the input lines during the transparent mode. For

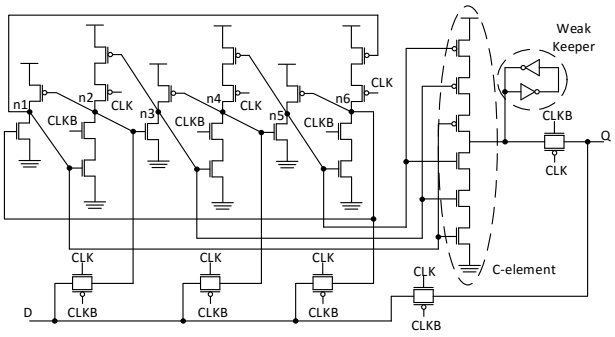

Fig. 1: HSMUF latch [11] with a weak keeper on the output.

example, if we observe node *n2* in Fig. 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. 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.

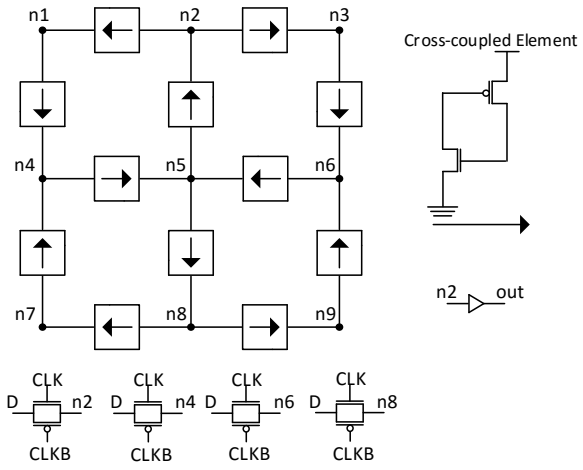

Fig. 2: DONUT latch as proposed in [13].

III. 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. 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

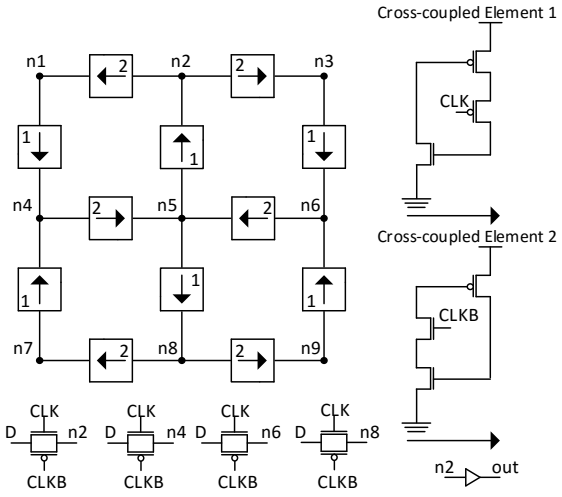

Fig. 3: Modified low-power DONUT latch.

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.

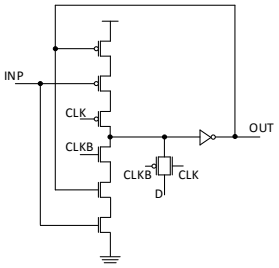

Fig. 4: Basic data storage loop block.

Using the basic storage block we construct the block based latch as in Fig. 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. 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. 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.

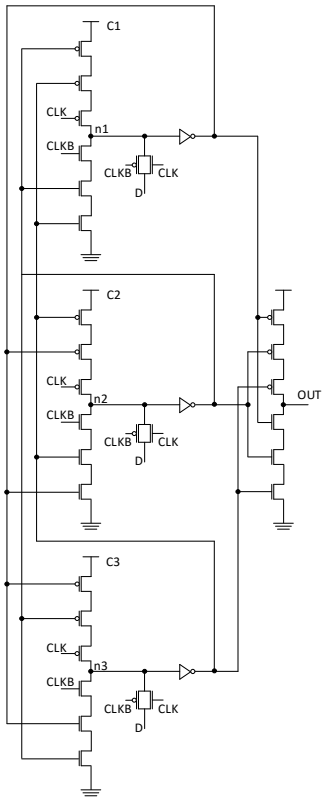

Fig. 5: Schematic of the block-based latch.

A problem, however, with the latch in Fig. 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. 7 we provide the schematic of the proposed HRDNUT latch. The design uses the block-based latch in Fig. 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. 6 provides the waveforms of the CLK, D and OUT nodes for both the transparent and hold modes of operation.

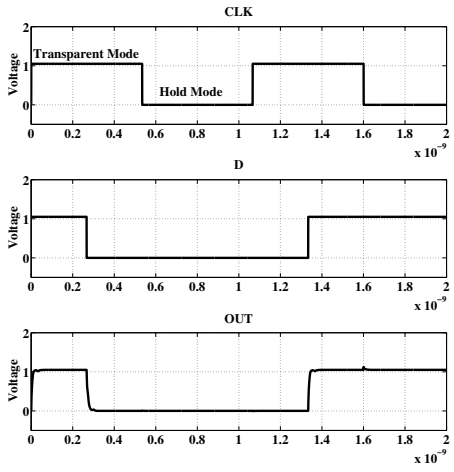

Fig. 6: Waveforms of the HRDNUT latch during normal operation.

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, 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*).
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

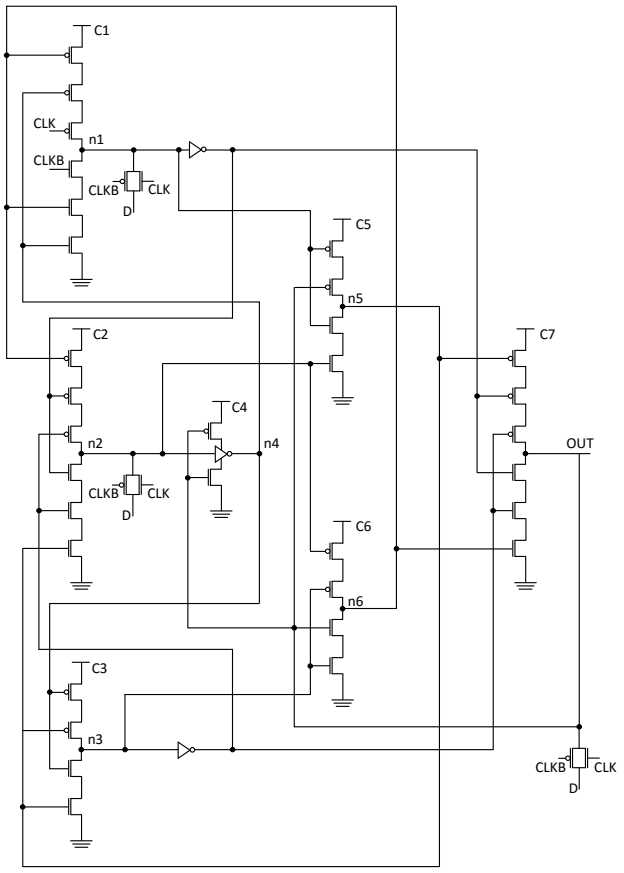

Fig. 7: Schematic of the HRDNUT latch.

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.

IV. Simulation Results

The proposed HRDNUT latch has been implement using the 1.05V 32nm PTM library [14] 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 [15]. 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.

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

Using equation (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 Figs. 8-17, we present the waveforms for each case discussed in Section III 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

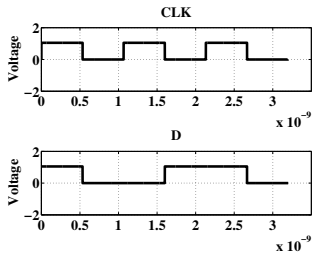

Fig. 8: Waveforms for CLK and D.

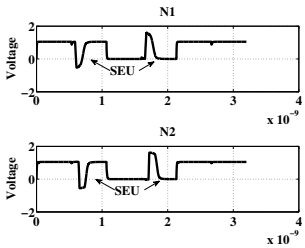

Fig. 9: Node pair n1 and n2 upset and recovery.

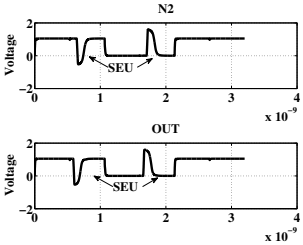

Fig. 10: Node pair n2 and out upset and recovery.

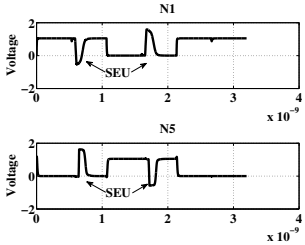

Fig. 11: Node pair n1 and n5 upset and recovery.

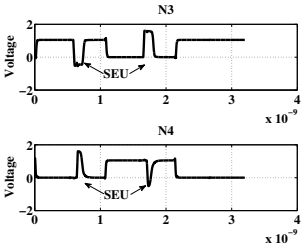

Fig. 12: Node pair n3 and n4 upset and recovery.

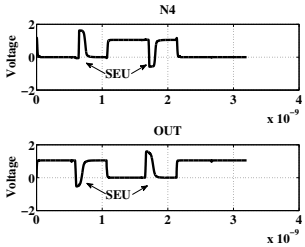

Fig. 13: Node pair n4 and out upset and recovery.

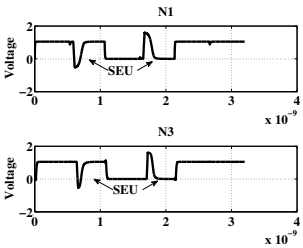

Fig. 14: Node pair n1 and n3 upset and recovery.

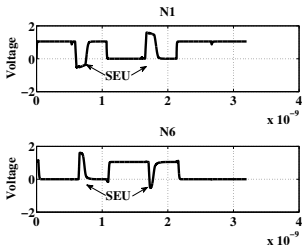

Fig. 15: Node pair n1 and n6 upset and recovery.

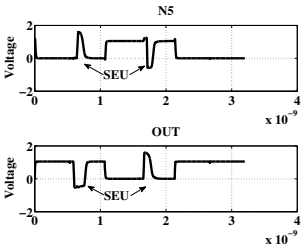

Fig. 16: Node pair n5 and out upset and recovery.

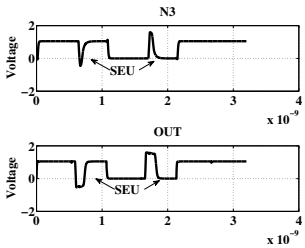

Fig. 17: Node pair n3 and out upset and recovery.

designed using the 32nm PTM library and operated at 1Ghz. For the analysis, we compare to the following SEU tolerant latches: DICE [1], FERST [3] and HIPER [2]. Additionally, we also compare to the following DNU tolerant designs: DNCS [9], Interception [10], HSMUF [11] 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 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 [9] which represents the number of unit sized (minimum width is W=40nm in this case) transistors required for the design. Table I provides the results of these simulations.

TABLE I: 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
[10]	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 [13]	Yes	Yes	4.021	14.722	54
DONUT-M (Section II)	Yes	Yes	2.760	8.421	72
HRDNUT (Proposed)	Yes	Yes	2.450	2.310	66

According to Table I 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 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.

V. Conclusion

In this paper we proposed the HRDNUT latch 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.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218126615500073

[6] 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)*, March 2007, pp. 273–278.

[7] 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.

[8] 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.

[9] K. Katsarou and Y. Tsiatouhas, "Soft error interception latch: double node charge sharing seu tolerant design," *Electronics Letters*, vol. 51, no. 4, pp. 330–332, 2015.

[10] ——, "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.

[11] 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.

[12] 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.

[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] 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

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