



**HAL**  
open science

# LDAVPM: A Latch Design and Algorithm-based Verification Protected against Multiple-Node-Upsets in Harsh Radiation Environments

Aibin Yan, Zhixing Li, Jie Cui, Zhengfeng Huang, Tianming Ni, Patrick Girard, Xiaoqing Wen

► **To cite this version:**

Aibin Yan, Zhixing Li, Jie Cui, Zhengfeng Huang, Tianming Ni, et al.. LDAVPM: A Latch Design and Algorithm-based Verification Protected against Multiple-Node-Upsets in Harsh Radiation Environments. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023, 42 (6), pp.2069-2073. 10.1109/TCAD.2022.3213212 . lirmm-03770056

**HAL Id: lirmm-03770056**

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-03770056v1>

Submitted on 6 Sep 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# LDAVPM: A Latch Design with Algorithm-based Verification Protected against Multiple-Node-Upsets in Harsh Radiation Environments

## ABSTRACT

In deep nano-scale and high-integration CMOS technologies, storage circuits have become increasingly sensitive to charge-sharing-induced multiple-node-upsets (MNUs) in harsh radiation environments. Muller C-elements are widely used for latch hardening against MNUs, such as triple and even quadruple node-upsets. Existing latch verifications for error-recovery highly rely on EDA tools with complex error-injection combinations. In this paper, a latch design with algorithm-based verification protected against MNUs in harsh radiation environments, is proposed. Due to the formed redundant feedback loops, the latch can completely recover from any MNU. Algorithm-based verification results demonstrate the MNU recovery of the proposed latch. Simulation results demonstrate the low area overhead of the proposed latch compared with the only one existing same-type design.

## KEYWORDS

Latch design, algorithm-based verification, fault tolerance, Muller C-element, multiple-node-upset

## 1 Introduction

With the aggressive reduction of transistor feature sizes, integrated circuits and systems can achieve high integration, low overhead and high performance. However, the sensitivity of CMOS devices to soft errors has significantly increased especially in harsh environments. When radiative particles, such as  $\alpha$  particles, heavy ions, and high energy neutrons, collide with sensitive nodes of integrated circuits, they may generate erroneous charges causing transient pulses or node-upsets that are called soft errors [1]. Soft errors include single node upsets (SNUs) and multiple node-upsets (MNUs). Double, triple, and even quadruple node-upsets (QNUs) are typical MNUs. Soft errors can severely affect the reliability of circuits and systems used in harsh radiative environments, e.g. aerospace [2]. Therefore, effective and scalable solutions for protection against soft errors are highly required.

To tolerate soft errors in the device level, many storage cells, such as latches [2-12], SRAMs [13-14], and flip-flops [15-16], have been proposed. This paper focuses on latch designs. For some traditional designs, a striking-particle can cause state changes of single nodes, resulting in SNUs. Due to charge-sharing [17], a striking-particle can cause state changes of multiple nodes, resulting in MNUs. Nowadays, due to the drastic reduction of transistor feature sizes, circuit integration is becoming higher and higher and thus node spacing is becoming smaller and smaller. Therefore, charge-sharing phenomenon is

becoming severer than ever before and a striking-particle can cause an MNU, such as triple-node-upset and even QNU [4].

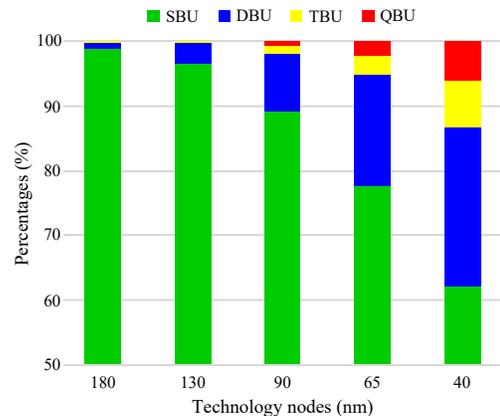


Figure 1: Single-, double-, and multiple-bit-upset percentages in different technology nodes for storage cells [18].

To the best of our knowledge, no research has indicated the occurrence probability of charge-sharing induced triple-node-upsets (TNU) and QNU. It is reported in [4] that providing an accurate and realistic calculation of the occurrence probability of a TNU/QNU is quite complex since many factors such as (1) technology data, (2) layout (to know effective area that may be affected by particles, spacing among adjacent nodes, etc.), (3) working conditions (hold mode duration, supply voltage, working temperature, etc.), (4) particle types (neutron, proton,  $\alpha$ -particle, heavy ion, etc.), (5) particle properties (flux distribution, effective hit rate, linear energy transfer, hit angle, etc.), and (6) particle correlations, etc., should be known.

However, it is reported in [18] that the aggressive reduction of transistor feature sizes can lead to triple bit-upsets (TBUs) and quadruple bit-upsets (QBUs). Figure 1 shows single bit upset (SBU), double bit-upset (DBU), TBU and QBU percentages in different technology nodes for storage cells. It can be seen from Fig. 1 that TBUs/QBUs are becoming a critical challenge as technology scales down. Therefore, in deep nano-scale and high-integration CMOS technologies, TNUs and QNUs, can also have an increasing occurrence probability, especially for circuits and systems used in harsh environments, seriously affecting the reliability of storage circuits.

On January 3, 2019, it was reported that “Chinese space craft is first to land on far side of moon”. The far side of the moon was relatively unexplored. The temperature in the far side of moon is considered to be quite low. To significantly reduce power dissipation, a latch can be switched in standby mode. In this case, the hold mode duration of the latch need to last for a long time. During this long duration, the latch can be impacted by a series

of radiative particles in harsh environments, thus causing a series of MNUs, such as TNUs and QNUs. Some may think that DNU-recovery is sufficient. However, if a DNU-recoverable latch suffers from a TNU, the DNU recoverability will be destroyed. Consequently, if the latch suffers from an SNU, the TNU along with the SNU can form a QNU. All of the above issues motivate us to design a highly reliable latch to tolerate MNUs, such as TNUs and QNUs, for safety-critical applications in harsh radiation environments.

In this paper, a Latch Design with Algorithm-based Verification Protected against MNUs in harsh radiation environments (LDAVPM) is for the first time proposed. The LDAVPM latch comprises two interlocked groups of 2-input Muller C-elements (MCEs) and five transmission gates. Each group can form a square so that two interlocked squares can be constructed to make the latch structure like a chip. Because the latch uses multiple levels of MCEs to form redundant and interlocked feedback loops, it can efficiently self-recover from any QNU so that it can self-recover from any TNU, DNU and SNU as well. Moreover, a high-speed path between the input and the output of the latch as well as the clock-gating technique are used to reduce overhead. For the algorithm-based verification, we firstly model the latch as a matrix and then analyze the relationship among all of the nodes of the latch when considering a QNU. Through the proposed algorithm, the directly and indirectly impacted nodes can be found and thus the related sub-matrix can be obtained. Since all sub-matrix do not contain a feedback loop to store the QNU-induced error, the latch can recover from any QNU. Simulation results demonstrate low-area and moderate delay overhead of the proposed LDAVPM latch.

The rest of the paper is organized as follows. Section 2 introduces the proposed solutions. Section 3 presents comprehensive comparison results for the proposed and existing state-of-the-art latches. Section 4 concludes the paper.

## 2 Proposed Solutions

### 2.1 Existing Component

MCEs are widely used for design-for-reliability. An MCE can output an inverted value if its inputs have the same value. If its inputs change and have different values, its output can temporarily have the previous correct value due to parasitic capacitance. Figure 2 shows the schematics of MCEs including a 2-input MCE and a clock-gating based 2-input MCE. A 2-input MCE has four important properties that are introduced as follows.

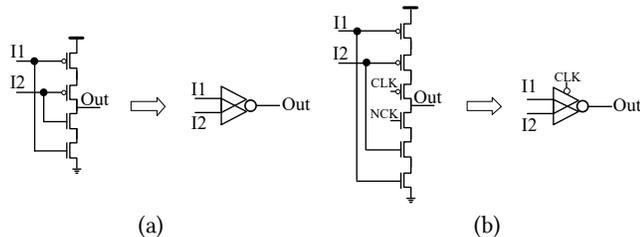


Figure 2: Schematics of Muller C-elements. (a) 2-input, and (b) Clock-gating based 2-input.

(1) **Recovery Property:** If all inputs of an MCE are correct, no matter whether its output is impacted or not, its output will provide the correct value.

(2) **Valid-retention Property:** If one input of an MCE is impacted but its output is not impacted, it will provide the correct output value, i.e., the error is masked.

(3) **Corruption Property:** If all inputs of an MCE are affected, it will provide a flipped output value. At this time, the inputs need recovery.

(4) **Invalid-retention Property:** If at least one input along with the output are simultaneously impacted, the output will keep the flipped value. At this time, the inputs need recovery.

### 2.2 Proposed Chip-like MNU-recovery Latch

Figure 3 shows the schematic of the proposed chip-like LDAVPM latch design protected against MNUs based on MCEs. It can be seen from Fig. 3 that the latch comprises two interlocked squares and five switches in the bottom. The outer square is constructed from eleven MCEs and an inverter, and the inner square is constructed from eleven MCEs and seven inverters. The inverters are used to ensure correct logic among MCEs. Note that the output of MCE13 is reused as the output (i.e., Q) of the latch. The switches are transmission gates, in which the gate-terminals of NMOS transistors are connected to the clock (CLK) signal and the gate-terminals of PMOS transistors are connected to the negative clock (NCK) signal, respectively. In the latch, D is the input, N1 through N22 are the internal nodes, CLK is the system clock signal, and NCK is the negative system clock signal, respectively. Figures 4 and 5 show the layout and MCE-output-topology of the proposed LDAVPM latch, respectively.

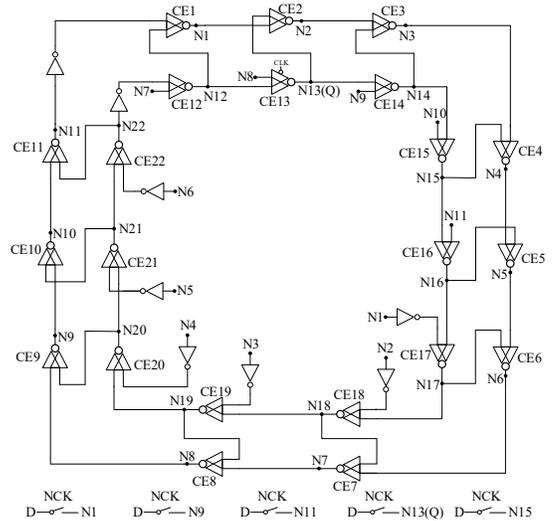


Figure 3: Schematic of the proposed chip-like LDAVPM latch design protected against MNUs.

When  $CLK = 1$  and  $NCK = 0$ , the latch works in transparent mode and all the transistors in the switches are ON so that N1, N9, N11, N13 (Q) and N15 can be pre-charged by D through the switches. It can be seen from figures 3 and 5 that N1 and N13 feed N2 (through MCE2) and thus the value of N2 can be determined.

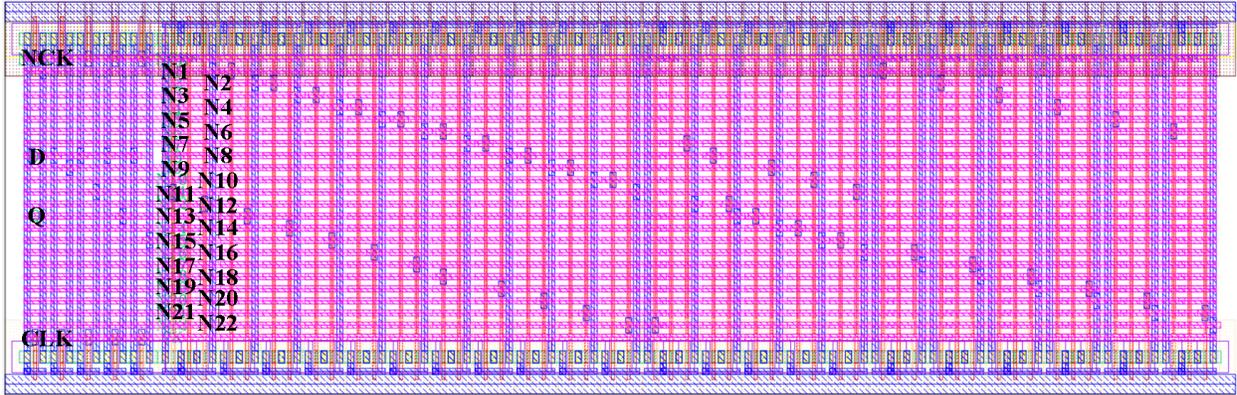


Figure 4: Layout of the proposed LDAVPM latch.

Similarly, the values of N14 and N16 can be determined. Then, the outputs of other MCEs can be subsequently determined (see Fig. 5). Therefore, all nodes of the proposed LDAVPM latch can be correctly pre-charged and Q can be determined by D in transparent mode. Node that clock-gating has been used in MCE13 to avoid current competition on Q (N13) to reduce power dissipation and transmission delay. This means that Q can be driven by D only through a switch but cannot be driven through MCE13 in transparent mode.

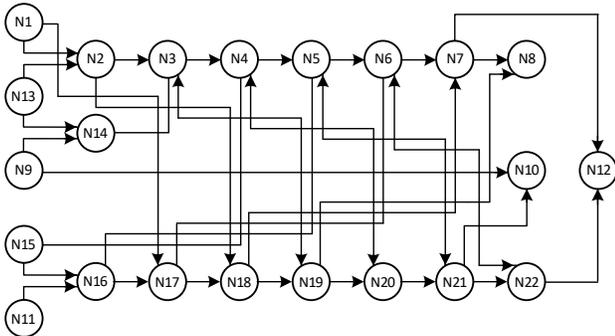


Figure 5: Topology of MCE-outputs of the proposed LDAVPM latch.

When  $CLK = 0$  and  $NCK = 1$ , the latch works in hold mode and all the transistors in the switches are OFF so that Q can only be driven through MCE13 instead of a switch. Since the output of an MCE in the outer square feeds a single input of an MCE in the inner square, and the output of an MCE in the inner square feeds a single input of the MCE in the outer square, many feedback loops (e.g.,  $N1 \rightarrow N11 \rightarrow N12 \rightarrow N1 \rightarrow N2 \rightarrow N3 \rightarrow N4 \rightarrow N5 \rightarrow N6 \rightarrow N7$ , etc.) can be constructed to store values for the latch. Therefore, the proposed LDAVPM latch can properly store values and can output the stored values through Q in hold mode.

### 2.3 Proposed Algorithm-based Verification Methodology

To verify MNU-recovery of the proposed latch, we for the first time model the latch as a matrix. Note that the input of an inverter in the latch is just the output of an MCE, and if the input of the

inverter suffers from an error, the output of the inverter will receive the error immediately. Therefore, we only need to consider the inputs of inverters (i.e., outputs of MCEs) for latch modeling.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
13	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
14	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
17	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
18	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
19	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
22	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0

Figure 6: Modeled matrix for the proposed LDAVPM latch.

Figure 6 shows the modeled matrix for the proposed LDAVPM latch. The matrix has 22 rows and 22 columns because the latch comprises 22 MCEs. The values in the matrix mean the connections among MCEs. “1” means “connected” and “0” means “disconnected”. For example, in the 1<sup>st</sup> row of the matrix, there are two “1” located in the 2<sup>nd</sup> and 17<sup>th</sup> positions but other values are “0”. This indicates that N1 only feeds the single inputs of MCE2 and MCE17 (see Fig. 3). Note that N1 needs to pass through an inverter to ensure the correct logic before feeding the single input of MCE17. Therefore, in the  $i$  row of the matrix, there are two “1” located in the  $j_1$  and  $j_2$  positions but other values are “0” ( $1 \leq i, j_1, j_2 \leq 22$ ). This indicates that  $N_i$  only feeds the single inputs of MCE $j_1$  and MCE $j_2$ . On the other hand, in the 1<sup>st</sup> column of the matrix, there are two “1” located in the 11<sup>st</sup> and 12<sup>nd</sup> positions but other values are “0”. This indicates that MCE1 only has two inputs N11 and N12 (see Fig. 3). Note that N11 needs to pass through an inverter to ensure the correct logic. Therefore, in the  $j$  column of the matrix, there are two “1” located in the  $i_1$  and  $i_2$  positions but other values are “0” ( $1 \leq j, i_1, i_2 \leq 22$ ). This indicates that MCE $j$  only has two inputs  $N_{i_1}$  and  $N_{i_2}$ .

Next, to verify the MNU recovery of the proposed latch, a novel verification algorithm is proposed (see Alg. 1). The algorithm has

three inputs, i.e., latch structure in a matrix form, all-node list of the latch, and flipped node count to indicate how many single nodes are simultaneously impacted to mimic an MNU. First, all feedback loops (circles) in the latch matrix can be found so that all nodes in the circles can be found. If the found nodes cannot contain all nodes of the latch, one or more nodes are isolated so that it is not a good latch. Otherwise, to mimic all possible MNUs, we select all-possible combinations of nodes from all-node list of the latch.

---

**Algorithm 1:** MNU Recovery Verification for a Latch.

---

**Input:** Latch-structure *Matrix*, All-node List *Node\_List*, Flipped Node Count *Flip\_Node\_Count*

**Output:** (see the print function)

```

1: Loops  $\leftarrow$  Find all possible feedback loops in Matrix
2: Nodes  $\leftarrow$  Find all nodes in Loops
3: if Nodes contains each node in Node_List then
4:  Test_lists  $\leftarrow$  Select all-possible Flip_Node_Count number of
   nodes from Node_List
5:  for each list in Test_lists do
6:    real_list  $\leftarrow$  list
7:    do
8:      is_add_node_to_real_list  $\leftarrow$  false
9:      for each node in Node_List but not in real_list do
10:       if real_list contains all inputs of an MCE whose
        output is node then
11:         add node to real_list
12:         is_add_node_to_real_list  $\leftarrow$  true
13:       end if
14:     end for
15:   while (is_add_node_to_real_list)
16:   sub_matrix  $\leftarrow$  Find sub_matrix that only contains each
   node in real_list from Matrix
17:   if sub_matrix has a feedback loop then
18:     print ("Error kept, i.e., the latch cannot recover.")
19:     break;
20:   end if
21: end for
22: else // At least one node is isolated
23:  print ("This is not a good latch.")
24: end if

```

---

For each combination to mimic an MNU, we create a list, i.e., *real\_list*, to remember each node in the combination. Then, we do the following steps (see Lines 7 to 15) until *real\_list* is not be updated. For each node *node* in the latch but not in *real\_list*, if *real\_list* contains all inputs of an MCE whose output is *node*, add *node* to *real\_list*. This means that if all inputs of an MCE have errors, the output of the MCE will receive the error immediately so that the output should be added into *real\_list*. As a result, we can get the final *real\_list* in which all nodes can be directly or indirectly impacted by the MNU. Finally, we find a sub-matrix *sub\_matrix* that only contains each node in the final *real\_list* from *Matrix*. If *sub\_matrix* has a feedback loop, the MNU error can be kept in the loop so that the latch cannot recover from the MNU and the algorithm can be terminated. If *sub\_matrix* does not have a feedback loop, we check the next combination to mimic a new

MNU. Therefore, if any *sub\_matrix* does not have any feedback loop, any MNU error cannot be kept so that the latch is recoverable from any possible MNU.

For the latch matrix in Fig. 6, all feedback loops of the latch can be calculated and it can be found that all nodes in all the feedback loops contain each node of the latch, i.e., no node is isolated from the latch. The scenario to verify QNU-recovery is discussed. For example, let us assume that node list <N1, N13, N14, N15> suffers from a QNU so that the initial value of *real\_list* is <N1, N13, N14, N15>.

For each node, e.g., N2, in the latch but not in *real\_list*, it can be seen from the latch matrix/structure that the inputs of MCE2 whose output is N2 are N1 and N13. Indeed, node list <N1, N13, N14, N15> contains N1 and N13. This means that all inputs of MCE2 are impacted so that the output of MCE2 will immediately impacted. Thus, output N2 needs to be added into *real\_list*. Through the above steps, other nodes/MCEs are also checked and we can get the new *real\_list* = <N1, N13, N14, N15, N2>.

For each node, e.g., N3, in the latch but not in the new *real\_list*, it can be seen from the latch matrix/structure that the inputs of MCE3 whose output is N3 are N2 and N14. Indeed, the new node list <N1, N13, N14, N15, N2> contains N2 and N14. This means that all inputs of MCE3 are impacted so that the output of MCE3 will immediately impacted. Thus, output N3 needs to be added into *real\_list*. Through the above steps, other nodes/MCEs are also checked and we can get the new *real\_list* = <N1, N13, N14, N15, N2, N3>.

It can be calculated that, for the QNU on <N1, N13, N14, N15>, through the above steps, we can obtain the final *real\_list* = <N1, N13, N14, N15, N2, N3, N4>. Therefore, we can extract the sub-matrix (see Fig. 7-(a)) that only contains each node in the final *real\_list* from the latch matrix. Indeed, it can be calculated that the sub-matrix does not have a feedback loop (see Fig. 7-(b)), so that the QNU-error cannot be kept. Thus, the latch can recover from the QNU on <N1, N13, N14, N15>. Then, we check the next node-combination to mimic a new QNU. It is found that, any sub-matrix does not have any feedback loop, i.e., any QNU-error cannot be kept. Therefore, the latch is recoverable from any possible QNU.

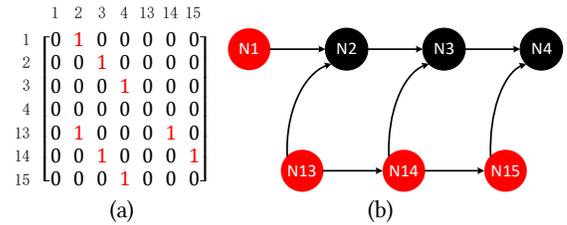


Figure 7: Sub-matrix and topology of related nodes for the proposed LDAVPM latch considering the final *real\_list* = <N1, N13, N14, N15, N2, N3, N4> = <N1, N2, N3, N4, N13, N14, N15>.

A counter-example is introduced here, in which five nodes should be considered. Let us assume that node-list <N3, N4, N8, N14, N18> suffers from errors. Then, the *real\_list* = <N3, N4, N8, N14, N18, N9, N19, N20> can be calculated and the sub-matrix can be constructed. Hence, the topology of related nodes can be obtained as shown in Fig. 8. In Fig. 8, the red nodes are directly impacted and the error can propagate to the black nodes. It can be seen from Fig. 8 that, at least one feedback loop, e.g., N3  $\rightarrow$  N19  $\rightarrow$

N20 → N9 → N14 → N3, can be formed, so that the errors can be kept through the feedback loop. Therefore, the proposed LDAVPM latch is not quintuple-node-upset recoverable.

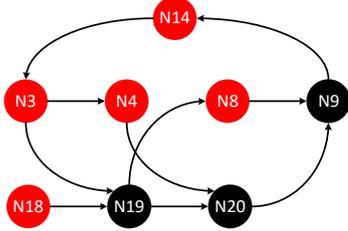


Figure 8: A counter-example for the proposed LDAVPM latch but considering five node-upsets.

Note that, for any single, double or triple node-upset errors, we can draw the same conclusion that the latch is recoverable from the errors. However, for quintuple node-upset errors and beyond, there is at least one feedback loop to keep the error. In summary, the proposed LDAVPM latch can recover from all possible single, double, triple and quadruple node-upset errors.

## 2.4 Error-free Simulation

The LDAVPM latch design was implemented in an advanced and commercial 22nm CMOS technology from GlobalFoundries and extensive simulations using Synopsys HSPICE were performed. In the simulation, the standard supply voltage was set to 0.8V, the working temperature was set to the room temperature, the PMOS transistors had the ratio  $W/L = 90\text{nm}/22\text{nm}$ , and the NMOS transistors had the ratio  $W/L = 45\text{nm}/22\text{nm}$ .

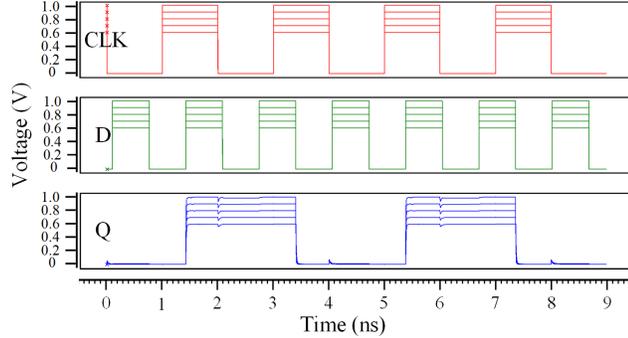


Figure 9: The error-free simulation results for the proposed LDAVPM latch.

Figure 9 shows the error-free simulation results for the proposed LDAVPM latch. It can be seen from Fig. 9 that the latch can propagate the value of input D to the output Q in transparent mode (i.e., when  $\text{CLK} = 1, D = Q$ ) and the latch can store the sampled D-value in hold mode (i.e., when  $\text{CLK} = 0, Q = D^*$ , where  $D^*$  is the latest D-value before the latch is entering into hold mode). Therefore, the proposed LDAVPM latch design can work correctly.

## 3 Evaluation and Comparison Results

To make a fair comparison, typical latch designs and the proposed LDAVPM latch were implemented with the same parameters listed in the previous sub-section (22 nm CMOS technology from GlobalFoundries, 0.8V supply voltage and the room temperature).

Table 1 shows the reliability and overhead comparisons among the SNU, DNU, TNU and/or QNU hardened latch designs. Note that “Tol.” denotes “Tolerant”, “Rec.” denotes “Recoverable” and “HIS Ins.” denotes “high-impedance-state (HIS) insensitive”, respectively. A latch suffering from node-upsets probably cannot self-recover but it can still output a correct value so that it is node-upset-tolerant (note that, when the latch subsequently suffers from another node-upset, the upsets can be accumulated to be an MNU in the worst case). However, a node-upset-recoverable latch suffering from upsets can remove the upsets so that it is called a node-upset-recoverable latch. Note that, if the inputs of an MCE become different for a long period of time, the output of the MCE will enter into HIS floating to an undetermined value so that node-upset-recovery is highly required. Generally, if an MCE is used as an output-level device of a latch and the latch cannot provide good node-upset-recovery, the latch is sensitive to the HIS.

It can be seen from Table 1 that, the TMR latch is only SNU-tolerant, but the LCQNUSR latch can additionally provide SNU recovery. Although the authors of the paper in [9] claimed the QNU recovery of their proposed LCQNUSR latch, comprehensive validations demonstrate that the latch cannot even tolerate a DNU (e.g., a DNU on node pair  $\langle N1, N3 \rangle$ ). The DNURL latch is not only SNU/DNU-tolerant but also SNU/DNU-resilient, but it cannot tolerate TNUs. These latches are not HIS-sensitive because they either do not use MCEs or can provide node-upset recovery. Note that the MNU-recoverable TNURL and QRHIL latches are also not HIS-sensitive. However, the other latches use an output-level MCE to output the stored value so that they are sensitive to the HIS. It can be seen from Table 1 that the TNURL latch can provide SNU/DNU/TNU tolerance and recoverability, but the HRHIL and the proposed LDAVPM latches can additionally provide QNU tolerance and recoverability. Therefore, the bottom two latches in Table 1 are more reliable (but the proposed LDAVPM latch has small silicon area and moderate delay as it will be discussed below).

Table 1 also shows the overhead comparisons among the SNU, DNU, TNU and/or QNU hardened latch designs. In Table 1, “Area” denotes the silicon area that is calculated based on extracted layouts, “D-Q Delay” denotes D to Q transmission delay (i.e., the average of rise and fall delays from D to Q), “CLK-Q Delay” denotes the average delays (rise and fall) from CLK to Q, and “Power” denotes the average power dissipation (dynamic and static), respectively.

In terms of silicon area, it can be seen from Table 1 that, the TMR latch consumes the smallest area but it is only SNU-tolerant. The LCQNUSR latch design is aimed to recover from QNUs so that its area is large. The TNUTL latch is the first solution to provide TNU-tolerance and its overhead is not optimized so that its area is large. However, the TNUHL latch can provide TNU-tolerance with optimized area. The TNURL latch is the first solution to provide TNU-recovery and its overhead is not optimized so that its area is large. The 4NUHL, QNUTL and QNUTL-CG latches consume moderate area, but they are not QNU-recoverable. To provide QNU-recovery, extra area is needed so that the QRHIL and the proposed LDAVPM latches consume extra area. However, compared to the QRHIL latch, the proposed LDAVPM latch consumes less area.

**TABLE 1: Reliability and Overhead Comparisons among the SNU, DNU, TNU and/or QNU Hardened Latches.**

Latch	SNU Tol.	SNU Rec.	DNU Tol.	DNU Rec.	TNU Tol.	TNU Rec.	QNU Tol.	QNU Rec.	HIS Ins.	Area ( $\mu\text{m}^2$ )	D-Q Delay (ps)	CLK-Q Delay (ps)	Power ( $\mu\text{W}$ )
TMR	YES	NO	YES	4.46	35.79	35.79	1.52						
LCQNUSR [9]	YES	YES	NO	NO	NO	NO	NO	NO	YES	12.47	2.41	2.42	1.24
DNURL [6]	YES	YES	YES	YES	NO	NO	NO	NO	YES	9.80	3.12	3.12	1.18
TNUHL [11]	YES	YES	YES	NO	YES	NO	NO	NO	NO	5.35	1.63	1.65	2.08
TNUTL [3]	YES	YES	YES	NO	YES	NO	NO	NO	NO	12.18	97.85	97.51	1.35
TNURL [2]	YES	YES	YES	YES	YES	YES	NO	NO	YES	19.01	5.44	5.47	1.18
DICE4TNU [5]	YES	YES	YES	YES	YES	NO	NO	NO	NO	7.72	1.63	1.63	2.20
4NUHL [10]	YES	YES	YES	NO	YES	NO	YES	NO	NO	10.99	2.54	2.42	1.29
QNUTL [4]	YES	YES	YES	NO	YES	NO	YES	NO	NO	9.50	1.89	1.96	1.62
QNUTL-CG [4]	YES	YES	YES	NO	YES	NO	YES	NO	NO	11.29	1.89	1.97	1.26
QRHIL [12]	YES	19.31	2.93	2.95	1.95								
LDAVPM (proposed)	YES	16.93	2.96	2.98	2.62								

In terms of D-Q delay, for the TMR and TNUTL latches, their delay is large due to the use of many logic devices from D to Q. In contrast, the other latches as well as the proposed LDAVPM latch have a small D-Q delay since they use high-speed transmission paths from D to Q. Note that, although most of the latches use a high-speed path from D to Q to reduce delay and their delay is similar, their delay is not the same. This is because, the input of a latch also needs to drive other different devices. It can be seen from Table 1 that the CLK-Q delay is close to the D-Q delay since they are related to the used devices from D to Q. For the proposed LDAVPM latch and many existing hardened latches, their CLK-Q delay is small since their D-Q delay is small.

In terms of power, among the latches in Table 1, the DNURL and TNURL latches consume the smallest power. Although their area differs significantly, compared to the DNURL latch, the TNURL latch has less feedback loops as well as less current-competition so that the large-area TNURL latch consumes the same power. The TNU-tolerant TNUHL and DICE4TNU latches have large power dissipation mainly because the feedback loops inside their structures are always active leading to more current-competition-induced power consumption. Compared to the same-type and QNU-recoverable QRHIL latch, the proposed LDAVPM latch consumes large power. This is because, the LDAVPM has much redundant feedback loops to provide QNU-recovery and the feedback loops are always active. Therefore, the proposed latch can provide error-recovery in real-time. In summary, the proposed LDAVPM latch can provide QNU recovery (and thus TNU, DNU and SNU recovery) for applications in harsh radiation environments with low area but at the cost of extra power.

## 4 Conclusions

CMOS technology scaling and charge-sharing lead to the occurrence of soft errors (e.g., TNUs and QNUs) in harsh radiation environments. Existing latches suffer from severe limitations such as non-recovery from QNUs, and/or large area. Moreover, existing recovery-validation highly relies on EDA tools. To address these issues, this paper has proposed a novel chip-like MCE-based and robust latch design recoverable from QNUs for applications in harsh radiation environments with algorithm-based QNU-recovery validation. Experiment results have demonstrated the

QNU recovery and low area overhead for the proposed LDAVPM latch.

## REFERENCES

- [1] M. Ebara, K. Yamada, et al, "Process Dependence of Soft Errors Induced by  $\alpha$  Particles, Heavy Ions, and High Energy Neutrons on Flip Flops in FDSOI," *IEEE Journal of the Electron Devices Society*, vol. 7, no. 1, pp. 817-824, 2019
- [2] A. Yan, X. Feng, Y. Hu, et al, "Design of a Triple-Node-Upset Self-Recoverable Latch for Aerospace Applications in Harsh Radiation Environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, pp. 1163-1171, 2020
- [3] A. Watkins and S. Tragoudas, "Radiation Hardened Latch Designs for Double and Triple Node Upsets," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 616-626, 2020
- [4] A. Yan, Z. Xu, X. Feng, et al, "Novel Quadruple-Node-Upset-Tolerant Latch Designs with Optimized Overhead for Reliable Computing in Harsh Radiation Environments," *IEEE Transactions on Emerging Topics in Computing*, early access, pp. 1-11, 2020
- [5] D. Lin, Y. Xu, X. Li, et al, "A Novel Self-recoverable and Triple Nodes Upset Resilience DICE Latch," *IEICE Electronics Express*, vol. 15, no. 19, pp. 1-9, 2018
- [6] A. Yan, Z. Huang, M. Yi, et al, "Double-Node-Upset-Resilient Latch Design for Nanoscale CMOS Technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 6, pp. 1978-1982, 2017
- [7] N. Eftaxiopoulos, N. Axelos, et al, "Delta DICE: A Double Node Upset Resilient Latch," *International Midwest Symposium on Circuits and Systems*, pp. 1-4, 2015.
- [8] A. Yan, C. Lai, Y. Zhang, et al, "Novel Low Cost, Double-and-Triple-Node-Upset-Tolerant Latch Designs for Nano-Scale CMOS," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 520-533, 2021
- [9] S. Cai, C. Xie, et al, "A Low-Cost Quadruple-Node-Upset Self-Recoverable Latch Design," *IEEE International Test Conference in Asia*, pp. 1-5, 2021
- [10] A. Yan, L. Ding, et al, "TPDICE and SIM based 4-Node-Upset Completely Hardened Latch Design for Highly Robust Computing in Harsh Radiation," *IEEE International Symposium on Circuits and Systems*, pp. 1-5, 2021
- [11] C. Kumar and B. Anand, "A Highly Reliable and Energy-Efficient Triple-Node-Upset-Tolerant Latch Design," *IEEE Transactions on Nuclear Science*, vol. 66, no. 10, pp. 2196-2206, 2019
- [12] A. Yan, A. Cao, Z. Fan, et al, "A 4NU-Recoverable and HIS-Insensitive Latch Design for Highly Robust Computing in Harsh Radiation Environments," *Great Lakes Symposium on VLSI*, pp. 301-306, 2021
- [13] C. Peng, J. Huang, C. Liu, et al, "Radiation-Hardened 14T SRAM Bitcell with Speed and Power Optimized for Space Application," *IEEE Transactions Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 2, pp. 407-415, 2019
- [14] Y. Chien and J. Wang, "A 0.2 V 32-Kb 10T SRAM With 41 nW Standby Power for IoT Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 8, pp. 2443-2454, 2018
- [15] S. Kumar, M. Cho, L. Everson, et al, "Analysis of Neutron-Induced Multibit-Upset Clusters in a 14-nm Flip-Flop Array," *IEEE Transactions on Nuclear Science*, vol. 66, no. 6, pp. 918-925, 2019
- [16] D. Lin and C. Wen, "DAD-FF: Hardening Designs by Delay-Adjustable D-Flip-Flop for Soft-Error-Rate Reduction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 1030-1042, 2020
- [17] J. Black, P. Dodd and K. Warren, "Physics of Multiple-Node Charge Collection and Impacts on Single-Event Characterization and Soft Error Rate Prediction," *IEEE Transactions Nuclear Science*, vol. 60, no. 3, pp. 1836-1851, 2013
- [18] A. Dixit and A. Wood, "The Impact of New Technology on Soft Error Rates," *IEEE International Reliability Physics Symposium*, pp. 486-492, 2011