A New EDAC Technique against Soft Errors based on Pulse Detectors

O. Ruano, P. Reviriego, J.A. Maestro
Dept. Ingeniería Informática-Universidad Antonio de Nebrija
Madrid, Spain
Email: {oruano, previrie, jmaestro}@nebrija.es

***Abstract*— In this paper, a new technique is proposed in order to assure the reliability of digital circuits against Single Event Upsets (SEUs) and Multi-Bit Upsets (MBUs), which are a major concern in a radiation environment like space. This proposal reduces the area cost of Triple Modular Redundancy (TMR), offering a similar protection level. In order to show the reliability and the area savings for this technique, it has been implemented using a commercial library to protect registers of different size. A software fault injection platform has been used in order to verify the reliability of the proposed technique.**

I. Introduction

Alpha particles, released by radioactive impurities and neutrons coming from outer space, are known to cause errors in digital devices [1][2]. Single and Multi-Bit Upsets may arise when these particles hit the storage components such as flip-flops and latches [3][4]. These errors do not cause permanent failures in hardware, and they are known as soft errors in the literature. In recent years, several cases of cosmic rays which induce soft errors in systems, have been reported [5][6]. Therefore, in the development of critical applications, designers have to include fault tolerant structures [7][8]. A commonly known method for SEU mitigation is Triple Modular Redundancy with voting, which performs a triplication of the target flip-flop, and adds a majority voter, in order to determine the right output of the circuit. If one of the modules is hit by an SEU, the voter will mask the fault using the free-error outputs (see Fig. 1).

In fact, this method is completely immune to a single upset. However, multiple simultaneous upsets would cause a failure unless the replicated bits are positioned physically distant from each other during the place and route phase. This technique increases the area and power consumption by a factor of three. Another general technique that can be applied to most digital circuits is the so-called Functional Triple Modular Redundancy [9] (FTMR, which also triplicates the combinational logic). Therefore, many studies propose new approaches that can improve the features of TMR. One example is Selective TMR [10], where the authors propose an optimization in the use of TMR, identifying the sensitive sub-circuits through a system based on signal probabilities. The TMR structures will be applied in the critical areas identified. For circuits with a high percentage of sensitive circuits, the hardened design approximates a full TMR implementation.

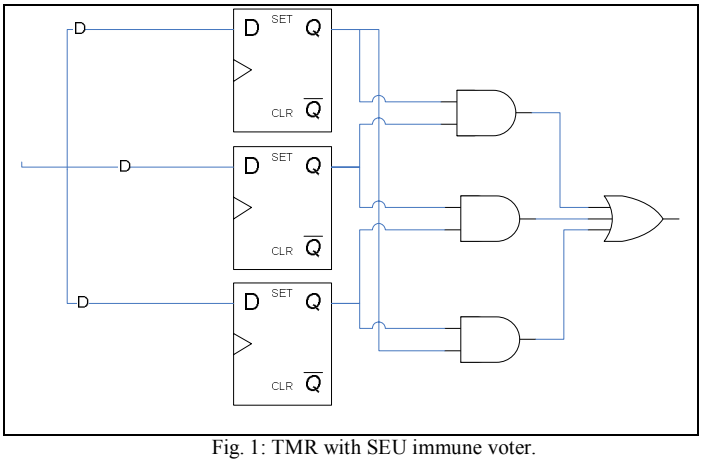

Fig. 1: TMR with SEU immune voter.

In [11], the authors propose two new redundancy techniques called Space-Time TMR (ST-TMR) and Enhanced ST-TMR (EST-TMR) which improve fault tolerance of both combinational circuits and sequential circuits respectively. The overhead and fault tolerance of both implementations are compared with conventional TMR showing a similar efficiency. In [12], the authors make a comparison of area and performance between Hamming codes and TMR.

In the rest of the paper a Pulse Detector based technique is presented as an alternative to TMR.

This paper is organized as follows. In Section II, the proposed approach is described in detail. A case study and results are introduced in Section III. Finally conclusions are presented in Section IV.

II. Description of the Proposed Technique

The purpose of this new technique is to supply the designers with fault tolerant circuits. An alternative solution that may be more efficient in terms of area and power than Triple Modular Redundancy and that can be used in order to protect digital circuits against SEUs assuring a level of reliability similar to TMR.

Before explaining in detail the basis of the approach, we are going to consider the characteristics required from the flip-flops so that they can perform adequately in a synchronous sequential

This work was supported by the Spanish Ministry of Education and Science under Grant ESP-2006-04163.

design and that are necessary for the proposed detection and correction (EDAC) system. Flip-flops will have one data input, one data output, one clock input and also, two special inputs which are provided to set or reset the flip-flop in an asynchronous manner, called direct (asynchronous) preset and direct (asynchronous) clear. It is in general unsafe to use these direct inputs for the normal operation of the synchronous system, but in this case, in order to achieve the protection targets, they are used only for a special event as it will be explained later, such as an SEU occurrence, and also for initialization.

In an edge-triggered flip-flop, the load of the state into the flip-flop is not caused (triggered) by the value of the clock signal but by its transition (edge). Since there is only one edge (leading or trailing) per clock cycle which can change the flip-flop state, the proposal to detect SEUs in the system is to take advantage of it, assuring a mode of operation that tolerates at most one change of state per clock cycle. Also this transition must coincide with the clock edge as it is showed in Fig. 2 and 3. Any other transition out of this allowed time frame will be an SEU. Note that the system must recognise when the change is caused by a reset condition. As the reset is asynchronous, it can produce a transition out of the allowed time frame mentioned before and therefore, be confused with an SEU.

D	1
Reset	1
Preset	1
CLK	0
Enable	1
Q	1

Fig. 2: Positive edge-triggering: allowed time.

D	1
Reset	1
Preset	1
CLK	0
Enable	1
Q	0

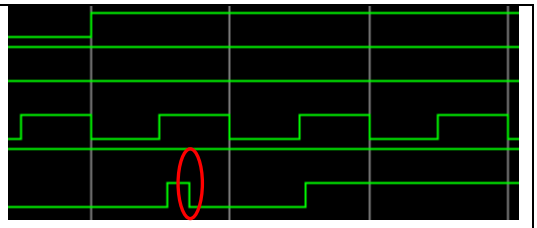

Fig. 3: Positive edge-triggering: not allowed time.

In order to develop this idea, a device capable of detecting edges, or in other words, a pulse detector is needed (see Fig. 4).

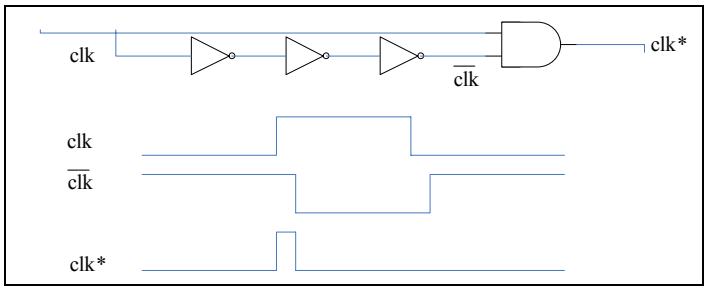

Fig. 4: Pulse detector.

At this point, the system is able to detect SEUs, based on the main idea introduced in the paragraphs before. Edges at the output of the flip-flops have to be detected, and if it has occurred at the same time that an edge in the clock signal, having into account the delay of the flip-flop in order to synchronize both signals. Now, in order to implement this function, a first pulse detector at the output of the D flip-flop that can detect the transition in signal Q (positive and negative transitions), and a second pulse detector whose input is the clock signal of the system in order to detect the allowed time frame where the flip-flops of the system can change their states can be used. If an edge is detected in signal Q and there is not an edge in the CLOCK signal, then it is an SEU. The scheme can be implemented as it is shown in Fig. 5:

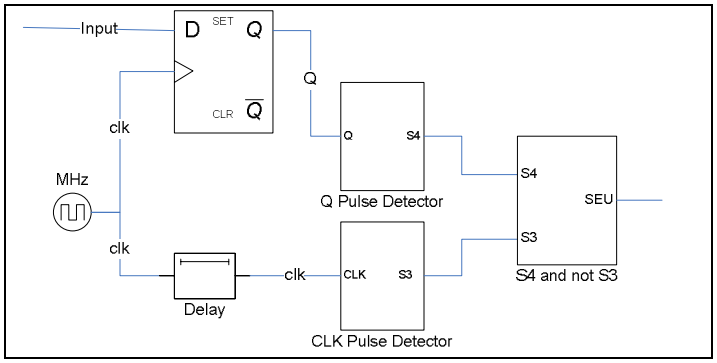

Fig. 5: EDAC Detection phase.

In this case, if the detection stage detects an SEU according to the previous scheme, the system has to change the current state of the flip-flop (it must invert the stored value). In order to achieve this, the output of the detection stage which indicates if there is an SEU can be used, as an input to a demultiplexer following the scheme showed in Fig. 6:

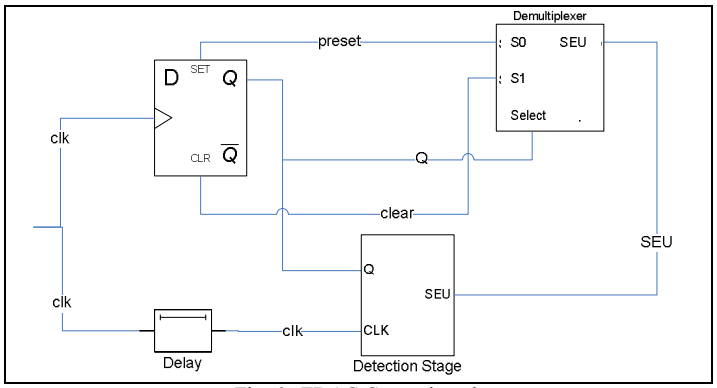

Fig. 6: EDAC Correction phase.

Besides this input, output Q acts as a select signal of the demultiplexer. Next, the two outputs of the demultiplexer are connected to the asynchronous inputs of the flip-flop in order to correct the current erroneous state. For instance, with this scheme if there is an SEU in the flip-flop, it is detected by the detection stage and if the current erroneous value of Q is zero, the system must correct the SEU inverting the value doing an asynchronous preset. On the other hand, if an SEU has been detected and the

value of Q is '1', the correction stage must invert the state doing an asynchronous clear to come back to the correct state. It must be noted that a correction in the system induces a positive or negative edge on Q which usually will not coincide with a clock edge and therefore, according to the functionality described before, the correction process would be confused with another SEU. In order to solve this issue, one option is to take advantage of the assumption that this technique is valid to deal with SEUs in the same way that TMR, which entails one bit flip per clock cycle only. To prevent this type of errors, an extra latch must be provided, which is used to keep the history of the current cycle in order to identify a previous SEU in the same cycle. So in this case this must not be considered as an SEU but a corrective action. About this additional register, we have to note that it will be controlled through two control inputs (set and reset). It will be reset each clock cycle through its reset and in the case of SEU, it will be changed to '1' through the set input. It is stimulated by the output of the pulse detector applied to the clock signal. If there is an SEU in a cycle, this latch stores a logic '1' preventing a new correction in the same cycle. An important note to emphasize about this element is that although it can be affected by an SEU during the normal function of the approach (from '0' to '1'), it will have no effect for the reliability of the system. Having a single SEU per clock cycle, if the latch suffers a bit flip, no recovery action will be carried out during this current cycle, but this process will not be necessary because no other SEU can occur in this cycle. In the beginning of the next clock cycle, it will be reset, deleting the SEU of the previous cycle. Now, for the current cycle, the system will be protected again. With this approach the structure of the EDAC is showed in Fig. 7:

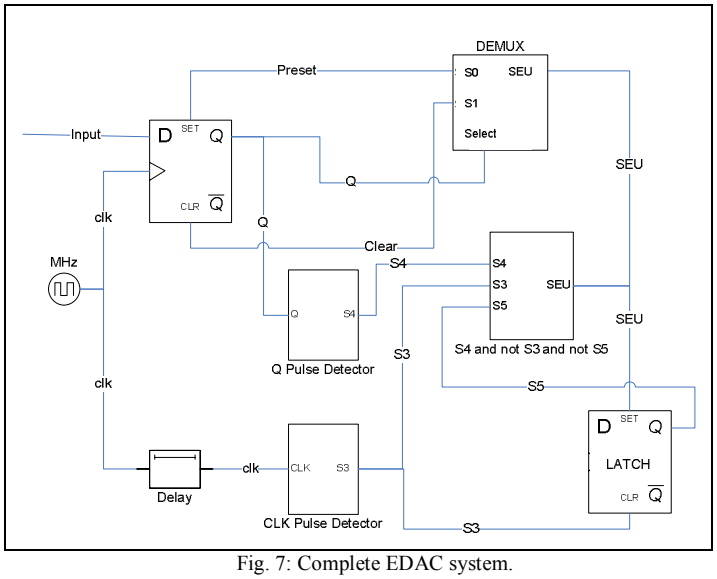

Fig. 7: Complete EDAC system.

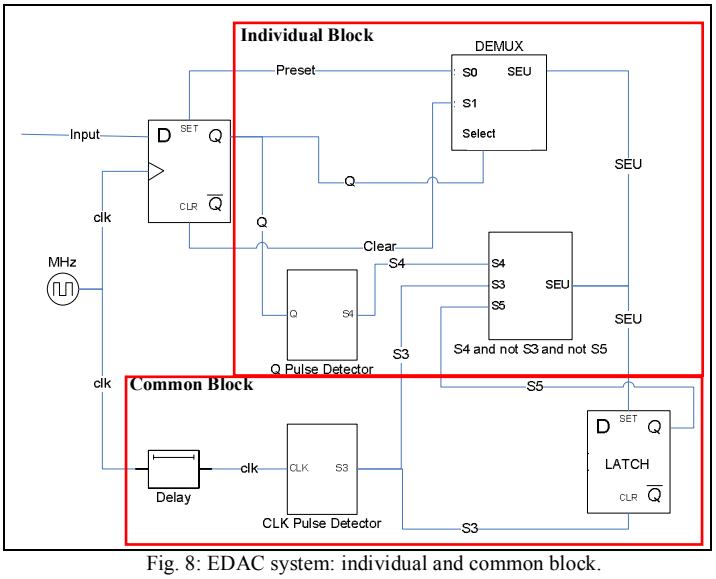

Fig. 8: EDAC system: individual and common block.

It must be noted that the approach can be applied to multiple registers. In the TMR technique, it is clear that any flip-flop needs to be replicated three times and also needs a particular majority voter. In our case, the complete system studied in the above scheme, does not need to be repeated for each flip-flop. The system can be divided in two blocks where the first of them must be implemented for each flip-flop of the design, and the second one can be shared among various flip-flops to get better area savings.

The first block contains the elements that must be implemented in each flip-flop. This block consists of a pulse detector for the Q output, a demultiplexer which controls the correction process, and a module which implements the SEU detector logic. This module triggers an SEU warning when it finds out that there is a Q edge and not a clock edge or a high level in the latch. This block is referenced in Fig. 8 in which it is labelled as individual block. On the other hand, the second block can be shared among some flip-flops of the system and it is composed of the pulse detector, which controls the transitions of the clock signal, and the latch that detects if there has been an SEU in any flip-flop during the cycle. This block can be common because the clock signal is the same for all components (flip-flops) of the system and the latch can be shared too, due to the assumption commented before (only one bit flip per cycle). If a flip-flop undergoes a change of its state in a predetermined cycle, the latch will store a logic '1' meaning that a given flip-flop accumulates an error and the individual first block of each state register will identify which has been affected.

This property makes the system scalable and it shows more advantages in terms of area cost compared with TMR when it is applied to a larger number of registers (in terms of number of flip-flops to protect). This factor will be analyzed in the next section where a comparison between our approach and the TMR technology is presented.

In summary, the EDAC system is composed of the two blocks shown in Fig. 8. It can protect a design which has N flip-flops, using N blocks of the individual type and only one block of the

common type. The structural design of a system with N blocks is illustrated in Fig. 9.

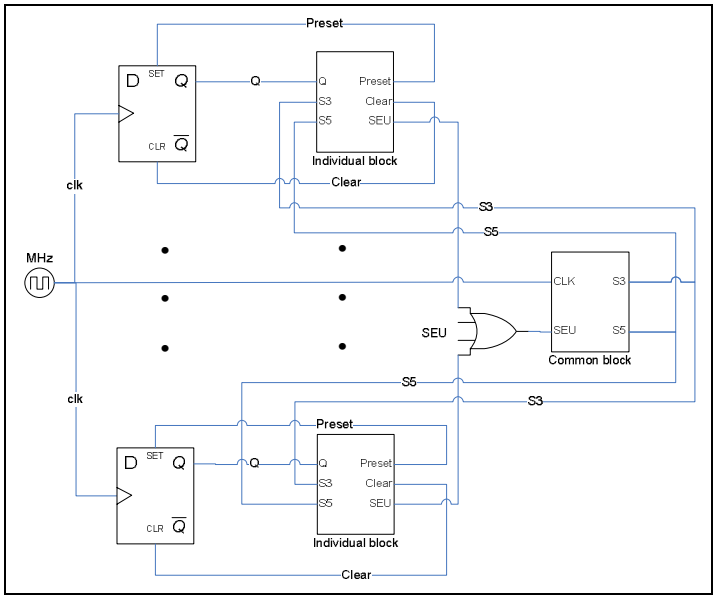

Fig. 9: EDAC system applied to a design with N flip-flops.

As it is showed, there are N individual blocks and only one common block. As a summary of this section, we will show an example of the explained functionality based on pulse detectors through some simulations where we illustrate the complete behaviour of the system:

D	1
CLK	0
Q	1
Pulse Detector CLK	0
Pulse Detector Q	0
LATCH output	0
SEU in Q	0

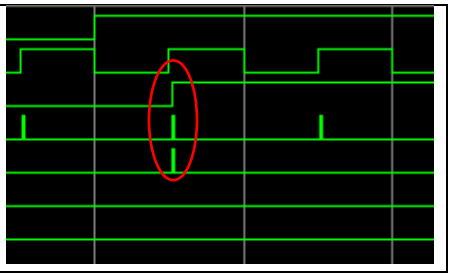

Fig. 10: Usual EDAC functionality: No SEU.

In this example (Fig. 10), it can be seen how the state of the flip-flop changes (Q) and the two detector edges coincide in the same time. So in this case the change is not considered an SEU. On the other hand in Fig. 11 the opposite case is showed:

D	1
CLK	1
Q	0
Pulse Detector CLK	0
Pulse Detector Q	1
LATCH output	0
SEU in Q	1

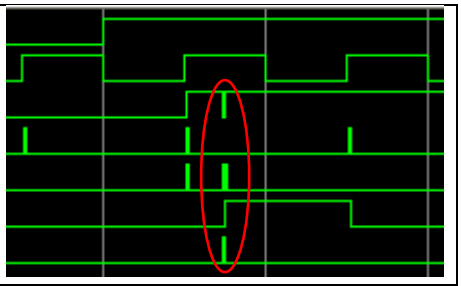

Fig. 11: Unusual EDAC functionality: SEU detected and corrected.

Now, Fig. 11 illustrates the appearance of an SEU in Q. The pulse detector of signal Q is activated but there is not an edge in the pulse detector of CLK, so it is an SEU. The signal SEU in Q is activated indicating that there is an SEU and the LATCH output maintains it during the cycle time. Any other change in the Q signal (state of the flip-flop) in the same clock cycle is not considered an SEU but a correction process.

III. Case Study and Simulation Results

Now that the properties of the detection and correction system have been explained in detail, the reliability performance of this structure as well as the benefits in terms of area using a 0,5 µm. commercial library (TSMC) will be analyzed. In order to prove the reliability and the advantages of our technique in terms of area compared with TMR, registers of different sizes (8, 16, etc.) composed of D type flip-flops, have been selected as a case study to obtain simulation results that verify the main conclusion of the previous section:

- The fault tolerant system based on pulse detectors offers a similar level of protection.
- The area cost for our approach is smaller than that of TMR.

First of all, we will show the area cost of the components that are needed to implement the solutions. In order to develop a TMR technique immune to SEUs (0% vulnerable) the scheme showed in Fig. 1 is implemented. For this design the area cost associated per flip-flop is (see Table I):

TABLE I
COSTS OF TMR APPROACH (IN AREA UNITS)

	TMR Method
Majority Voter	20.3 a.u
3*FFD	74.4 a.u.
Total Area	94.7 a.u.

On the other hand, the proposed technique based on pulse detectors needs the following components with their costs associated per flip-flop:

TABLE II
COSTS OF PULSE DETECTORS APPROACH (IN AREA UNITS)

	Pulse Detector Method	
Block (individual)	FFD = 24.8 Q Pulse detector = 11.9 a.u. Function = 13.2 a.u. Demultiplexer = 25.8 a.u.	
	Total area	75.7
Block (common)	Delay = 25.8 a.u. Clk pulse detector = 11.7 a.u. Latch = 26 a.u.	
	Total area	63.5
Total Area	139.2 u.a.	

For only one flip-flop the result is clearly favourable to the use of TMR. But this is an unrealistic case. Usually a design has a large number of flip-flops which form registers. So our priority is the

study that shows how the area increases with the number of flip-flops which require protection. An example of the advantages of our proposal can be observed using different sizes of register. The next table can give an idea of this improvement (Table III).

TABLE III
COMPARISON OF AREA COSTS (IN AREA UNITS)

	Pulse Detector Method	TMR Method	Savings
8-bit Register	689.4	757.6	9.9%
16-bit Register	1310.6	1515.2	15.61%
32-bit Register	2568.6	3030.4	17.97%
64-bit Register	5069.0	6060.8	19.56%
128-bit Register	10085.4	12121.6	20.18%
256-bit Register	20102.6	24243.2	20.59%
512-bit Register	40152.6	48486.4	20.75%
1024-bit Register	80237.0	96972.8	20.85%

With the results of Table III the improvements in the area performance of our proposal are captured. It is noticeable that the progress in terms of area difference stays virtually invariable as the improvement limit is reached (1).

$$\frac{n * TMR}{n * IB + CB} \cong \frac{TMR}{IB} \quad (1)$$

Where n is a variable that indicates the number of protected flip-flops; TMR indicates the area cost in order to protect the n flip-flops; IB and CB specify the area of the individual and common blocks respectively. If n is large, (1) converges to the limit showed.

In addition, this structure has an overhead in area and delay. As the number of flip-flops increases, more control logic has to be added (network of OR gates needed to route the n SEU signals) as shown in Fig. 9. Also, the delay increases and can suppose an issue for the functionality. In order to solve both difficulties, the way to implement this technique consists in clustering the design flip-flops in groups which will be protected with our solution individually. This has an advantage over TMR when dealing with MBUs, as we will explain later.

The next issue is to asses experimentally how the reliability performance is, compared to TMR.

For the next experiments a register of 8 bits has been chosen as case study. The same results are expected for other register sizes. These tests consist in applying the different protection methods to the register (pulse detector and TMR methods) and compare them. The expected results should show a high level of reliability in both cases. If it happens, it means that the technique can decrease the area costs keeping the reliability with respect to TMR. For these tests, a fault injection platform based on simulation that uses the Single Event Upset Simulation Tool (SST), has been used to make the setup and the injection of the SEU campaigns (see [13] [14] [15] for more details). In Table IV, the results obtained during the SEU campaigns are showed where the number of SEUs injected in the proposed technique and in TMR is 1000. The results in Table IV also illustrate the reliability in terms of SEU percentage which are not detected by the protection techniques.

TABLE IV
FAULT GRADING

Protection Technique	Injected SEUs	Vulnerable SEUs	
None	1000	1000	100%
TMR	1000	0	0%
Pulse detector	1000	12	1.2%

With these results, it can be noticed that the pulse detector solution provides similar level of protection against SEUs as the TMR solution. Comparing the efficiency and the area results obtained using software-based fault injection between the two solutions, the next points can be concluded:

- For the pulse detector technique, the register is 98,8% effective against SEUs, similar to TMR which achieves 100%.
- The pulse detector solution is more efficient in terms of area than TMR when it is applied to multiple bit registers.

On the other hand, the small percentage of vulnerability showed by our solution (around 1%) depends directly on only one factor. If the synchronization of the pulses is accurate the only instant where an SEU cannot be detected is when it coincides with the edge of the clk pulse detector. For instance, if we suppose a period for the clock of 100 ns and a pulse width of 1 ns for the detectors, only one nanosecond is vulnerable to SEUs per each cycle.

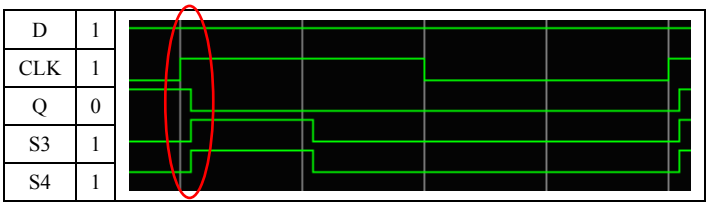

Fig. 12: Example of an undetected SEU.

Fig. 12 shows an SEU which coincides with the edge of the clock. While the value of D is equal to '1', the signal Q changes its value because of an SEU. As this transition (S4) coincides with the edge of the clock (S3), it is not considered an SEU, but the signal D confirms that it is an SEU. Only this kind of SEUs can cause an erroneous functionality.

It is important to ensure the action of synchronizing the response behaviours of the two elements of memory involved in the system. The response time of the latch must be smaller than the flip-flop because when an SEU is detected by the system, the correction process bit-flips the state of the flip-flop which contains the SEU and, as we anticipated in a previous section, the correction if the latch output has not been updated yet, will be considered as a new error, becoming an erroneous decision. So in this case, after the real SEU has been detected by the SEU detector logic, which has as inputs S4 equal to '1' and the rest equal to '0', the system corrects the SEU inverting the state of the flip-flop but one moment before, the state of the latch (S5)

should store the value '1'. This new latch state changes the stimulus of the SEU detector logic in order to reject a new correction because now the third term corresponding to the output (S5) is '1'. So the SEU detector logic (S4=1, S3=0, S5=1) shows in its output a '0' which means not an SEU.

Although this technique has been designed to avoid the effects of SEUs (one error per clock cycle), the previous assumption about the synchronization makes necessary to divide the design in groups that will be protected individually by our system protection due to the delay in the network of OR gates, which generates the input of the latch. It is interesting to note that this architecture is more appropriate than TMR to protect a circuit against MBUs. This is because the spatial relationship, among the flip-flops in this architecture, is smaller than in the case of TMR if the target flip-flops are clustered around different latches. The MBUs that impact in this architecture only can cause faults when they bit-flip at the same time both the latch and any registers that are grouped around this latch. Any other situation, like the MBU affecting several registers inside or outside the group, is irrelevant because they are corrected by the system. So in order to avoid MBUs, only the latch and the flip-flops associated need to be physically distant, while in TMR the three flip-flops associated to the voter should be separated. Therefore, the proposed technique reduces the effort necessary in the place and route of all flip-flops because our architecture has fewer flip-flops that need to be separated.

IV. Conclusions and Future Work

The benefits that this solution can offer versus TMR have been analyzed simulating the system on different registers, and injecting SEUs through a simulation tool developed by the ESA. Firstly, we have calculated the difference in terms of area costs versus TMR, and secondly the grade of reliability of this technique, obtaining the following conclusions:

- The approach reduces the area costs compared with the TMR solution.
- The efficiency is similar to the offered by TMR.
- This technique is more suitable than TMR to deal with MBUs.

The future work will be oriented to use this technique in more complex circuits and study in detail the advantages versus TMR. Another suggested study is the comparison with other techniques, like Hamming codes.

References

[1] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies", IEEE Transactions on Device and Materials Reliability, vol. 5, pp. 305-316, Sep 2005.

[2] R.D. Schrimpf and D.M. Fleetwood, "Radiation effects and soft errors in integrated circuits and electronic devices", World Scientific Publishing, ISBN: 981-238-940-7, 2004.

[3] P. E. Dodd and LL. Massengill, "Basic Mechanisms and Modelling of Single-Event Upset in Digital Microelectronics", IEEE Transactions on Nuclear Science, vol. 50, No 3, pp. 583-602, June 2003.

[4] D. Radaelli, H. Puchner, S. Wong and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device", IEEE Transactions on Nuclear Science, vol. 52, no. 6, pp. 2433-2437, 2005.

[5] S. E. Michalak, K. W. Harris, N. W. Hengartner, B. E. Takala and S. A. Wender, "Predicting the number of fatal soft errors in Los Alamos National Laboratory's ASC Q supercomputer", IEEE Transactions on Device and Materials Reliability, vol. 5, pp. 329-335, Sep 2005.

[6] A. Lesea, S. Drimer, J. J. Fabula, C. Carmichael and P. Alfke, "The Rosetta experiment: Atmospheric soft error rate testing in differing technology FPGAs", IEEE Transactions on Device and Materials Reliability, vol. 5, pp. 317-328, Sep 2005.

[7] B. W. Johnson, "Design and Analysis of Fault-Tolerant Digital Systems", Addison-Wesley, 1989.

[8] M. Nicolaidis, "Design for Soft Error Mitigation", IEEE Transactions on Device and Material Reliability, vol. 5, no. 3, pp. 405-418, September 2005.

[9] S. Hanbic, "FTMR:Functional Triple Modular Redundancy", ESA Report FPGA-003-001, December 2002.

[10] P. K. Samudrala, J. Ramos and S. Katkoori, "Selective triple modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs", IEEE Transactions on Nuclear Science, vol. 51, pp. 2957-2969, Oct 2004.

[11] W. Chen, R. Gong, K. Dai, F. Liu and Z. Wang, "Two New Space-Time Triple Modular Redundancy Techniques for Improving Fault Tolerance of Computer Systems", IEEE Computer and Information Technology CIT, pp. 175 – 175, Sept. 2006.

[12] R. Hentschke, F. Marques, F. Lima, L. Carro, A. Susin and R. Reis, "Analyzing area and performance penalty of protecting different digital modules with Hamming code and triple modular redundancy", Proc. of IEEE Integrated Circuits and Systems Design, pp. 95 – 100, Sept. 2002.

[13] D. Gonzalez, "Single Event Upset Simulation Tool Functional Description", ESA Report TEC-EDM/DCC-SST2, July 2004.

[14] O. Ruano, P. Reviriego, J.A. Maestro and P. Reyes, "A Simulation Platform for the Study of Soft Errors on Signal Processing Circuits through Software Fault Injection", Proc. of IEEE International Symposium on Industrial Electronics, 2007.

[15] Computer Architecture and Technology Group, Universidad Antonio de Nebrija, Web page. Available: http://www.nebrija.es/~jmaestro/esa