A Methodology for Automatic Insertion of Selective TMR in Digital Circuits Affected by SEUs

O. Ruano, J. A. Maestro, *Member, IEEE*, and P. Reviriego, *Member, IEEE*

***Abstract*—In this paper, a methodology to perform automatic selective TMR insertion on digital circuits is presented, having as a constraint the required reliability level. Such reliability is guaranteed while reducing the area compared to TMR. In addition, a performance enhancement is proposed in order to guarantee a computation time feasible for this automatic selective TMR insertion methodology. It focuses on the choice of a starting point close enough to an optimal solution. The method consists in the analysis of the topological features of the target circuit which will help the optimization engine to identify those flip-flops more susceptible to be tripled depending on the showed sensitivity to SEUs.**

***Index Terms*—Fault injection, optimization, single event upsets (SEUs), triple modular redundancy (TMR).**

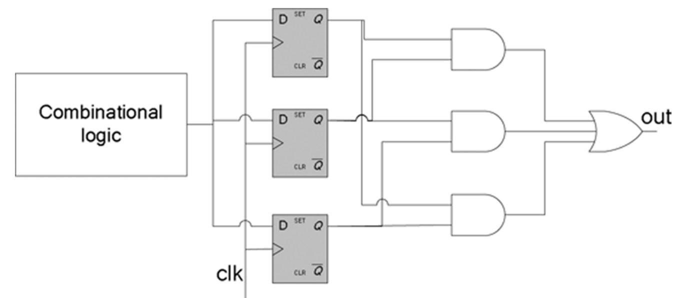

Fig. 1. TMR memory cell with single voter.

I. Introduction

FAULT tolerance on semiconductor devices has been a meaningful matter since upsets were first experienced in space applications several years ago. Integrated circuits operating in the space environment can be upset by charged particles that generate errors in the system. Therefore, the interest in studying fault-tolerant techniques in order to keep integrated circuits operational has increased. In order to guarantee the circuit reliability against Single Event Upsets (SEU), some mitigation techniques have been proposed in literature during the last few years. These techniques cover a wide range of methods which can be classified as fabrication process-based, design-based and recovery techniques. The fabrication process-based, such as Epitaxial CMOS processes [1] and Silicon-on-Insulator (SOI) [2], can reduce the radiation effects of Total Ionization Dose (TID) and Single Event Latch-up (SEL). However, they do not eliminate SEUs. The design-based techniques, also called architectural techniques, are based on logic redundancy such as Triple Modular Redundancy (TMR) [3], Error detection and correction coding (EDAC) like Hamming [4] or hardened memory cells like HIT and DICE [5]. Finally, new techniques based on recovery have been proposed for SRAM-Based FPGAs. The main idea is to recover the original programmed information after an upset [6]. Among all these methods, the well-known TMR has become a common practice. In this case, the SEU sensitive logic (memory cell) is tripled and voters are placed at the outputs to identify the correct value (Fig. 1). In [7], the TMR technique is shown to be a suitable solution to provide reliability against SEUs, for an 8051 microcontroller. The fault injection results showed that less than 1% of the bit upsets could provoke an error in the output of the TMR design. However, the TMR technique presents some overheads because of its full hardware redundancy, such as area and power dissipation.

As a result, both area and power consumption are increased, so many applications cannot accept this high cost.

In order to make the TMR technique valid for many applications, a heuristic method called selective TMR is proposed in [8], [9] to make a selective insertion of TMR in the nodes that present a bigger vulnerability to SEUs. This proposed method is based on the logic masking analysis of each node. In [10], [11], this technique is improved mixing the logic masking analysis with an innovative study called timing masking. It consists in computing the error latching window and its aim is to have equal delays to outputs from each gate in the circuit through gate relocation, in order to achieve a better solution. Another technique for soft error mitigation is presented in [12]. It is based on a selective addition of redundant wires that can prevent the distorted signal from propagating to an output or a storage element. In [13], partial TMR is applied to the circuit sections that affect structures which cause a persistent error. The rest of works found in literature propose new fault tolerant techniques that try to minimize the area cost. This is the case of the DWC technique introduced in [14] that combines time and hardware redundancy reducing the area compared to TMR. Other examples can be found in [15] and [16]. Focusing on the idea of the selective TMR, and on systems that can allow a fixed error rate, an analysis of an innovative method for selective TMR insertion is presented in this paper, which provides an automatic selective insertion based on an iterative optimization method. The problem is addressed as an optimization task where two variables must keep a balance: the reliability level demanded by the user must be assured in the final design, while, on the other

Manuscript received September 05, 2008; revised January 09, 2009. Current version published August 12, 2009. This work was supported by the Spanish Ministry of Science and Education under Grant ESP-2006-04163, the Regional Government of Madrid and the European Union FEDER programme.

The authors are with the Universidad Antonio de Nebrija, C/Pirineos, 55 E-28040 Madrid, Spain (e-mail: oruano@nebrija.es; jmaestro@nebrija.es; previrie@nebrija.es).

Digital Object Identifier 10.1109/TNS.2009.2014563

0018-9499/$26.00 © 2009 IEEE

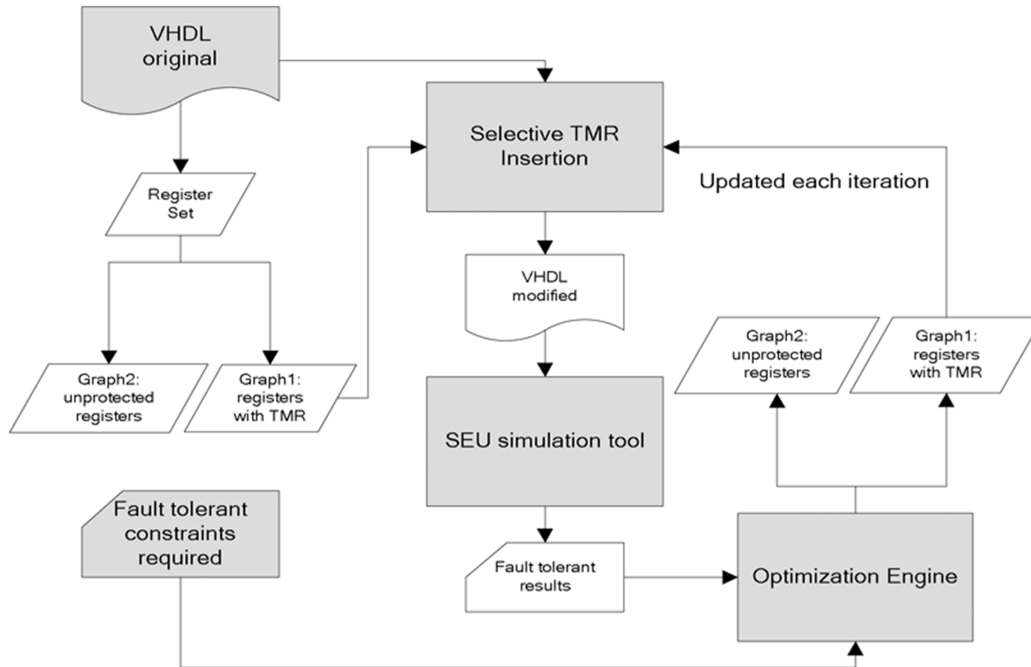

Fig. 2. Methodology workflow.

hand, the area cost due to the modular redundancy must be minimized. This paper is organized as follows. Section II introduces the proposed methodology for TMR optimization and explains in detail the major factors that are involved. In Section III, a tri-state pipeline stage is used as a case study where the methodology is validated by fault injection experiments in VHDL. In Section IV, the methodology is supplemented by an innovative static analysis process which enhances the performance of the optimization process. In Section V, the different criteria are analyzed experimentally via fault injection. Section VI shows in detail the previous case study with the enhancement integrated in the method. Final remarks and future work are placed in Section VII, followed by the references.

II. Proposed Methodology

Currently, the design of radiation tolerant circuits is very expensive in terms of area and power consumption if the Triple Modular Redundancy (TMR) technique is applied. Therefore, this work proposes a selective TMR insertion adapted to the fault tolerance level required by the application. Although TMR clearly provides a superior protection level, not all the applications demand the highest fault tolerance level. In some cases, an intermediate protection could be enough, what makes TMR excessive in terms of area cost. The proposed methodology is based on a combinatorial optimization problem which makes use of the reliability constraints that the target circuit has to meet. The issue is addressed like a graph partitioning problem which is included in the category of NP-complete problems. It consists in partitioning a given initial graph which includes all registers of the initial circuit, in two sub-graphs. One of them, stores the registers that the system has chosen to protect with TMR, and the second stores the remaining registers without TMR. The purpose of this methodology is to find the best partition that assures the minimum possible area in terms of the number of registers on which TMR is inserted, while meeting the reliability constrains specified for the circuit. Therefore, the optimization process looks for a solution among a finite set of alternative results which minimizes the area and assures the required reliability. The methodology workflow defined to drive the process is described in Fig. 2.

The operational method is detailed next. First of all, data inputs required by the methodology must include an original VHDL description, a random initial partition of the registers grouped into two sub-graphs and the fault tolerance constraints required by the application. Once these data are available, the methodology is able to obtain a valid partition which can satisfy the specified reliability constraints. The first step consists in measuring the fault tolerance for the random initial partition in order to evaluate it and see if it is a valid solution or not. In order to achieve this objective, the process is detailed next. The original VHDL and the set of registers with TMR are given to the Selective TMR insertion module in order to insert TMR on the appropriate registers. Next, the modified VHDL is used to perform a fault injection campaign using the SEU Simulation Tool [17], [18]. The fault tolerance level of the modified VHDL is reported by the fault injection results through software fault injection [19]. These results are sent to the optimization engine module which evaluates them against the reliability constraints required by the application. Once the initial partition has been measured, the above process is repeated with a new partition of the registers selected by the optimization engine, which manages the remaining possible partitions through an optimization algorithm. For each new partition driven by the optimization algorithm, the Selective TMR Insertion module modifies the original VHDL and a new fault injection is run to evaluate the new tolerance. The process is finished when the optimization engine

cannot find a new partition with less area cost that satisfies the demanded reliability. In the following, some issues which can have impact in the proposed methodology are discussed.

A. Optimization Algorithm

In order to guide the successive movements and the several partitions which are tested through the fault injection process, an iterative algorithm has been selected to perform the experimental results. Fiduccia-Mattheyses [20] is the meta-heuristic algorithm which has been chosen due to its widely use in literature. A final solution is achieved by moving one register at a time, for one graph of the partition to the other, updating the partition in each step. Another feature of this algorithm is that it reduces the chance that the minimization process becomes trapped at local minima (hill-climbing) [20]. In an attempt to reduce the number of registers with TMR while satisfying the reliability constraints, it drives the intermediate partitions according to a cost function which will determine if each partition can satisfy the reliability constraints or not.

B. Cost Function

The cost function must drive the algorithm in order to accept or deny each movement that is proposed by Fiduccia-Mattheyses in the search space. This cost function is included in the minimization problem in the following way.

- Given: a function $f\colon A \rightarrow R$.
- Sought: an element x_0 in A such that $f(x_0) \leq f(x)$ for all x in A.

The domain A of f is called the search space, while the elements of R are called candidate solutions or valid solutions. A valid solution that minimizes the objective function is called an optimal solution. With these premises, the selective TMR insertion issue can be characterized in the following way.

- A represents all possible partitions which can be generated from the whole set of registers that compose the design under test.
- x_0 represents a valid solution, i.e., a partition composed by two sub-graphs in which the first contains the selected registers to protect with TMR and the second the rest of registers without TMR that meets the specified reliability constraints.

All valid solutions must satisfy a specified set of constraints. In our case, a set of reliability constraints on the circuit outputs are used. Consequently, in order to measure the quality of a solution according to the parameters of the problem (number of registers, reliability constraints and real error rate obtained by the fault injection), the next cost function has been defined so that combines all these parameters

$$\begin{aligned}\mathrm{F} = {} & \#\mathrm{registers} + \mathrm{k}_1 \cdot \max(\#\mathrm{error}_1 - \mathrm{reliability_constraint}_1, 0) + \\ & + \mathrm{k}_2 \cdot \max(\#\mathrm{error}_2 - \mathrm{reliability_constraint}_2, 0) + \cdots + \\ & + \mathrm{k}_\mathrm{n} \cdot \max(\#\mathrm{error}_\mathrm{n} - \mathrm{reliability_constraint}_\mathrm{n}, 0)\end{aligned}$$

where *#registers* is the total number of register that composes the currently partition, error_i the number of errors detected in the i output through the fault injection process, $\mathrm{reliability_constraint}_i$ is the error threshold that must be satisfied by each output and finally the k_i coefficients, which represent the influence of each output. Currently, the cost function takes no account of the overhead of the majority voters in an explicit way because this factor is directly proportional to the number of tripled registers.

C. Reliability Constraints

Reliability constraints mentioned previously $(\mathrm{reliability_constraint}_i)$, are defined as the tolerance level which must be met by a valid solution in any case. These constraints are measured as the maximum percentage of wrong cycles allowed in each output of the target design. The optimization engine and more specifically the F cost function requires these parameters in order to accept or deny the movement of each register. These factors must be specified by the user as the tolerance thresholds which must be met by each output of the circuit and therefore, their choice is considered key for the optimization process. Depending on these constraints, the circuit can tolerate more or less failures, and as a result it will have a direct effect on the insertion of selective TMR.

D. Optimization Performance

It is known that the runtime required to obtain a valid solution, for the chosen iterative algorithm, increases with the size of the problem. In some cases, the computation time can be so high that the heuristic method is not able to obtain accurate solutions in reasonable runtimes (too many registers). According to this issue, it is detected that the initial partition (P_{initial}) has a major influence to reduce the execution time of the method as it will be shown in the experimental results of the next section. In this case, a previous knowledge or a static analysis which reports information of possible weak points of the circuit (Section IV) can improve the runtime performance. This would reduce the number of iterations, initiating the process from a selected initial partition close to a valid solution instead of a random initial partition. Also, from this information, the cost function can be customized through the k coefficients in order to prioritize those weak points identified as sensitive to SEUs, converging earlier to a solution. Finally, other kinds of algorithms can be used in order to make a performance comparison, like a greedy heuristic. Moreover, two heuristics can be studied to address the following two sources of complexity. The first one would consist of avoiding exhaustive simulation of all states combinations using sampling and the second one would be referred to reduce the search space, considering the addition of TMR on those flip-flops which reach a primary output with a high probability [21], [22].

For this work, all these points have been taken into account in the next section in order to present experimental results which optimize the insertion of TMR in a selective way (P_{initial}, reliability constraints and k coefficients).

III. Case Study

In order to verify the proposed methodology, a case study is analyzed in this section. The circuit under test computes the sum

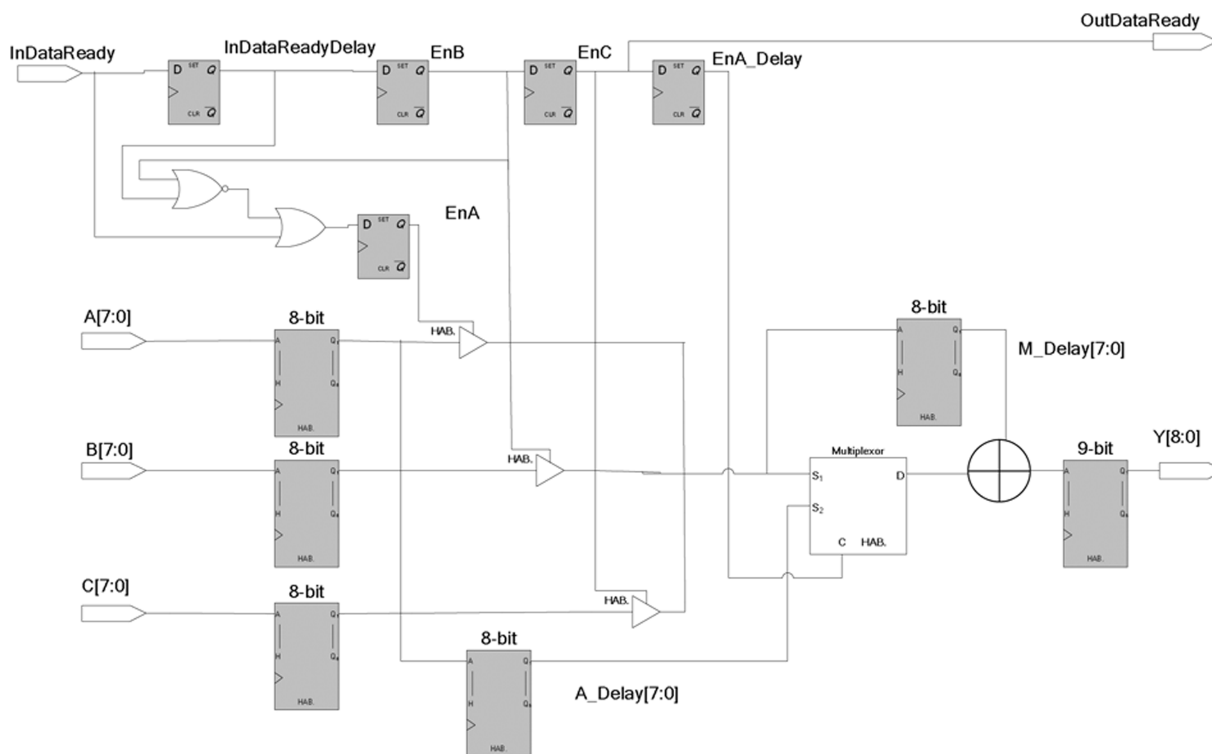

Fig. 3. Tri-state pipeline stage.

TABLE I
AUTOMATIC INSERTION OF SELECTIVE TMR FOR AREA EFFICIENCY

% Allowed wrong cycles	$P_{initial}$ (flip-flops)		Detected wrong cycles		Explored partial solutions	P_{final} (flip-flops)		Selective TMR Cost reduction
	TMR	Without TMR	Abs. Value	% Value		TMR	Without TMR	
Y= 0%	0	54	0	0%	103	54	0	162 (0%)
	26	28	0	0%	58	54	0	162 (0%)
	54	0	0	0%	12	54	0	162 (0%)
Y= 0,5%	0	54	450	0,45%	73	30	24	114 (29.6%)
	19	35	490	0,49%	28	30	24	114 (29.6%)
	54	0	493	0,49%	39	30	24	114 (29.6%)
Y= 1%	0	54	998	1%	71	21	33	96 (40.7%)
	12	42	985	0,98%	28	21	33	96 (40.7%)
	54	0	992	0,99%	46	21	33	96 (40.7%)

of three 8-bit data buses A, B, and C, that is to say $(A+B)$, $(B+C)$, and $(A+C)$ and supplies the results on a single 9-bit output bus in three consecutive clock cycles. Also, an extra output control signal (OutDataReady) should be active for one clock cycle to indicate when the three summed outputs are available. The proposed design (54 flip-flops) is shown in Fig. 3 [23].

The proposed methodology has been applied to this circuit in order to obtain results of the automatic insertion of selective TMR. Experimental results are collected in Table I, where three different scenarios regarding the requested tolerance (0, 0, 5, and 1 per cent respectively) have been proposed in order to show the area decrease compared to TMR. Also, for each reliability scenario, three different tests have been proposed where the initial partitions are changed in order to study how the method performance can vary in terms of the explored solutions. The reliability constraint has been considered only for the Y output as all registers of the system have effect on it, included those which determine the value of the OutDataReady output. All tests have run simulations of 100 000 clock cycles injecting random campaigns of 10 000 SEUs with a minimum separation of 3 cycles between two consecutive ones. These results allow a comparison with the area cost of the standard TMR approach (Table I).

With these results, it can be noticed that in the cases where the application does not demand the highest fault tolerance level (for example 0,5% or 1%), an intermediate protection could be enough to satisfy the reliability constraints, reducing the area cost in terms of flip-flops (29.6% and 40.7% respectively) compared to the standard TMR approach.

On the other hand, an interesting result offered by the methodology consists in the detection of critical nodes which show a bigger vulnerability to SEUs like in the case of the control flip-flops $(\mathrm{InDataReady_Delay}, \mathrm{Ena}, \mathrm{Enb}, \mathrm{Enc})$. In all tests, these registers have been tripled with a high priority in order to achieve the fault tolerance needed. In this way, the $\mathrm{P_{final}}$ illustrated in Table I for tolerances of 0, 5, and 1% has included in all cases these control registers in the TMR sub-graph, and in second place the $\mathrm{M_Delay}$ and Y registers were also added. Finally, it also should be noticed that the initial partition is a significant factor in order to improve the performance of the process. In this way, an innovative analysis based on the topology of the circuit will complete the optimization methodology in the next section, in order to select the best initial partition. This will reduce the number of searches required by the engine. Table I emphasizes that when beginning from a partition near to the final solution, a small number of searches is required. For example, in the case of 0% tolerance in order to achieve the same $\mathrm{P_{final}}$ (54-0), starting from a partition without any register tripled (0-54) 103 explorations are required while using a better partition, like (26-28) or the best (54-0), 58 and 12 are needed respectively, improving the performance considerably.

These results can be contrasted with an analysis of the circuit. As it has been said, in all final partitions the control flip-flops have been included. This means that these flip-flops are considered more sensitive to SEUs by the process. An SEU in these flip-flops can alter significantly the good behavior of the circuit being able to propagate the error during three cycles in the worst case (SEU on InDataReady flip-flop). The right functionality is only recovered when the SEU goes out of the delay line of the control path $(\mathrm{Ena_Delay})$. On the other hand, $\mathrm{A_Hold}, \mathrm{B_Hold}, \mathrm{C_Hold}$, and $\mathrm{A_Delay}$ data registers are considered less sensitive because they refresh the useful data each cycle and also are used $1/3$ of the cycles, so an SEU can be observed at the output or it cannot affect the output value. So, these registers usually have not been tripled by the system due to the low probability of observing an SEU in the Y output. Nevertheless, an SEU in the remaining registers $\mathrm{M_Delay}$ and $\mathrm{Y_Delay}$ will be observed in all cases but only during one cycle. With this analysis, it has been proved why the methodology firstly triplicates control registers, secondly $\mathrm{M_Delay}$ and Y, and finally, if it does not satisfy the reliability constraints yet, the rest of registers.

IV. Performance Enhancement: Selection of the Initial Partition

In the previous section, it could be noticed the importance of a good initial partition for a successful optimization process. This factor is used as an input for the optimization engine and it has been demonstrated that it has a direct influence on the performance of the methodology. Depending on the initial partition, the optimization process can reach the best solution exploring more or less partial solutions, as it is shown in Table I. So, as the computational time depends on the circuit size, this can cause an unacceptable computational time. In order to mitigate this weakness, an enhancement will be presented and studied in an experimental way, validating it and later integrating it into the workflow, as it is shown in Fig. 4.

In this figure, a new module has been included to improve the performance. It drives the selection process in order to choose the best starting point for the optimization engine. This method is based on an intuitive topological analysis of the target circuit, which is able to improve the computational time. The output of this module is a suggested initial partition which is supposed to be close enough to the final solution so that the computation time is feasible, reducing the effort of the optimization algorithm. In Fig. 5, this idea is illustrated. Given a search space, the goal of the proposed enhancement is not to look for the optimal solution (S) (this is the task of the optimization engine), but an approximation (D) which can reduce the number of steps needed to get the solution. With this, it is possible to partly reduce the dependence between the computational time and the size of the circuit.

As it has been said above, the way to obtain an initial partition close to the optimal solution is based on the topological analysis of the flip-flops in the circuit. In order to do this, some topological criteria have been chosen to study how they can influence the correct behavior of the circuit in the presence of SEUs. These criteria have been analyzed experimentally for several circuit samples, allowing to draw conclusions about their relation to SEU sensitiveness. In other words, by examining these criteria, it will be possible to identify those flip-flops which are 'hot spots' against SEUs. Therefore, the proposed approach allows labeling this group of flip-flops as candidates to be part of Graph 1 of the start partition (tripled registers) (Fig. 4).

In order to characterize each flip-flop according to the criteria, the circuit is considered as a finite directed graph G on n vertices (flip-flops) where its adjacency matrix is used to represent the connectivity of the different nodes (Fig. 6) [24].

The adjacency matrix is the $n \times n$ matrix where each no-diagonal entry $\mathrm{a_{ij}}$ is a direct connection from flip-flop i to flip-flop j, and each diagonal entry $\mathrm{a_{ii}}$ is the feedback or loop at flip-flop i. In the following, a set of topological criteria are chosen, allowing the characterization of each flip-flop. They have been described and finally analyzed, for a wide space of circuit samples, which allow estimating their relevance in a statistical way.

A. FAN-IN Input Parameter

This parameter refers to the number of flip-flops which have influence on the state of the analyzed flip-flop. The main reason to use this characteristic is the fact that the greater the number of inputs which drive the flip-flop state is, the less likely that an SEU can be stored in it. This is because the effect of the distorted value can be masked by the rest of inputs. On the other hand, if there is only one input that drives the target flip-flop and it is distorted, the SEU will be stored always. So, depending on the number of inputs of the implemented function, the flip-flop can be more or less tolerant to SEUs [8].

Algorithm 1 is implemented in order to measure this attribute on a given topology.

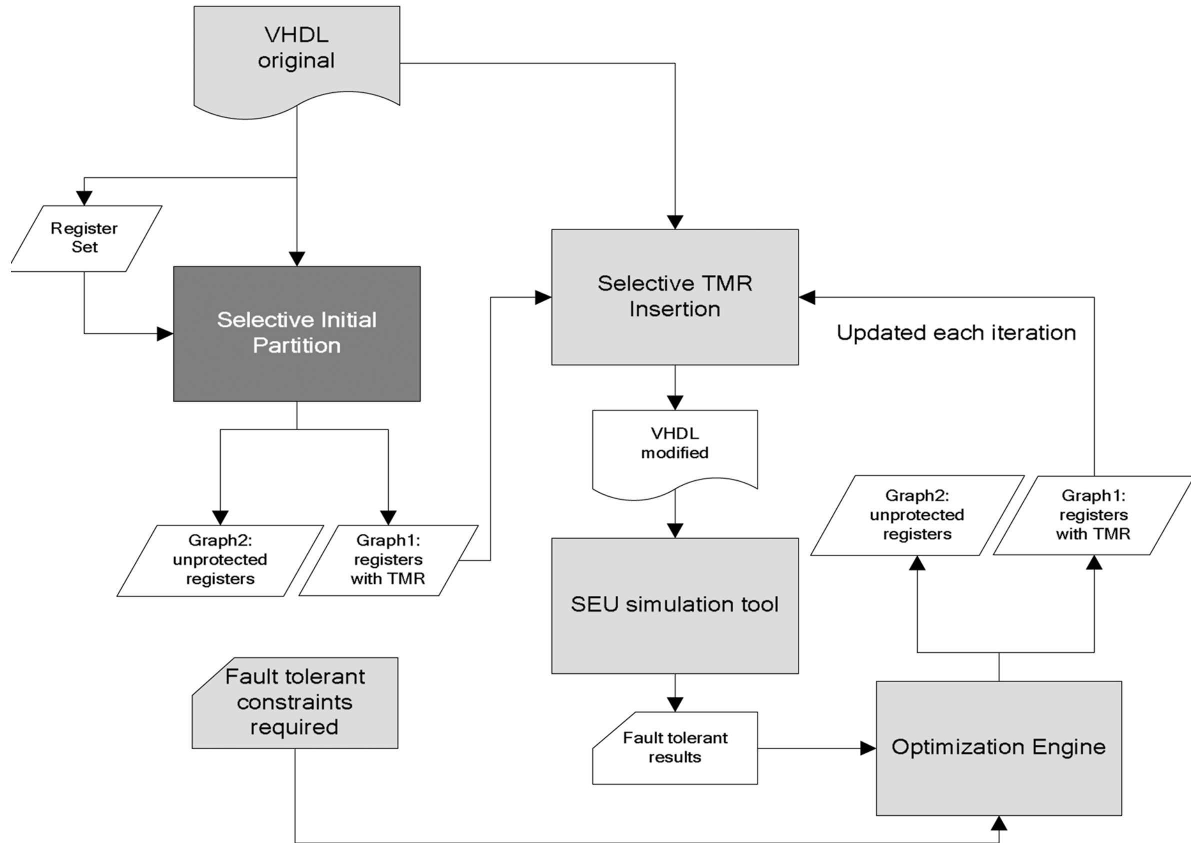

Fig. 4. Methodology workflow with the added enhancement.

Algorithm 1: input_criterion (Adjacency Matrix)

Function: input_criterion (Adjacency Matrix)
Input: Adjacency Matrix, $\mathrm{n \times n, n \in N}$

Foreach $\mathrm{r_i \in register_set}$ (R) do

{

$\mathrm{C1[r_i] = length\ (Find(:, r_i) = 1)}$

}

Return C1

B. FAN-OUT Output Parameter

This parameter refers to the number of flip-flops which are affected directly by the state of each analyzed flip-flop. The main reason to decide on this characteristic is because an SEU in a flip-flop which propagates its state to several paths has a high probability of affecting any design output, and therefore its observability is increased. Algorithm 2 is implemented in order to measure this attribute:

Algorithm 2: output_criterion (Adjacency Matrix)

Function: output_criterion (Adjacency Matrix)

Input: Adjacency Matrix, $\mathrm{n \times n, n \in N}$

Foreach $\mathrm{r_i \in register_set}$ (R) do

{

$\mathrm{C2[r_i] = length}$ (Find $\mathrm{(r_i, :) = 1)}$

}

Return C2

C. Influence on Output Paths

This parameter refers to the number of primary outputs which are affected by the analyzed flip-flop. The main reason to take this issue into account, is the fact that an SEU in a flip-flop belonging to more than one output path, has a higher probability to propagate the error to at least one of these outputs, increasing its observability. Algorithm 3 is implemented in order to measure this attribute, using the Dijkstra's algorithm, which is often used in computer networks [25], [26]. It is a graph search algorithm that solves the single-source shortest path problem for a graph. In this case Dijkstra's algorithm is used to find the number of outputs which are linked to each evaluated flip-flop. When Dijkstra returns a path between the flip-flop and the output, it means that there is connectivity.

Algorithm 3: Influence_output_criterion (Adjacency Matrix)

Function: Influence_output_criterion (Adjacency Matrix)

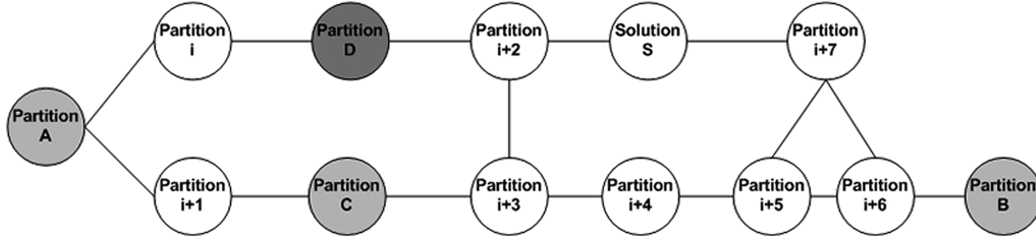

Fig. 5. Selecting the initial partition: partition (A) All_TMR partition; (B) None_TMR partition; (C) Random partition; (D) Based on topological analysis; S) Optimal Solution.

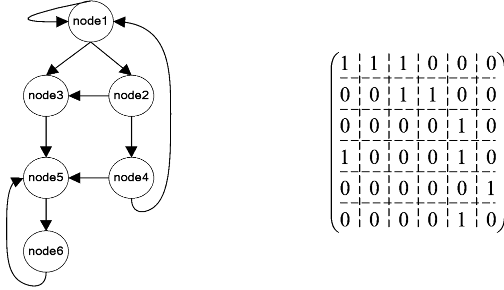

Fig. 6. Adjacency matrix example.

Input: Adjacency Matrix, $n \times n, n \in N$

Foreach $r_i \in$ registers_set (R) do

{

  $C3[r_i] = 0$

  Foreach $o_i \in$ output_set (O) do

    {

      If (dijkstra(Adjacency Matrix, $r_i, o_i) < \infty$

      $C3[r_i]$ ++

    }

}

  Return C3

D. Proximity to the Outputs

This parameter refers to the proximity of the evaluated flip-flop to the outputs which are affected by it. The main reason to take this criterion into account is the fact that depending on the distance to the output, an SEU can have a lower or higher probability to be masked. It is due to factors like masking logic, timing logic and so on [10], [11]. An SEU in a flip-flop with a greater distance to the output is more likely to be masked. Algorithm 4 is implemented in order to measure this attribute, using Dijkstra's algorithm. In this case, it is used to find the shortest path between each evaluated flip-flop and all other circuit outputs affected by them.

Algorithm 4: proximity_output_criterion (Adjacency Matrix)

Function: proximity_output_criterion (Adjacency Matrix)

Input: Adjacency Matrix, $n \times n, n \in N$

Foreach $r_i \in$ register_set (R) do

{

  $C4[r_i] = \infty$

  Foreach $o_j \in$ output_set (O) do

    {

      // take the minimum distance among all affected outputs

      $C4[r_i] = \mathrm{Min}(\mathrm{dijkstra}(\text{Adjacency Matrix}, r_i, o_j), C4[r_i])$

    }

}

  Return C4

E. Feedback Loops

This parameter refers to the situation in which the output signal of a flip-flop is passed (fed back) to its input, directly or through several previous stages. This is often used to control the dynamic behavior of the system. The main reason to take this criterion into account is the fact that an SEU in this type of flip-flop (with feedback) can be stored in the loop, making the effect of the SEU permanent. Algorithm 5 is implemented in order to measure this attribute, using the same Dijkstra algorithm. In this case, the direct outputs of the target flip-flop are identified and from each one, Dijkstra algorithm looks for the shortest path to the target flip-flop. If this path is found, it means that there is a loop.

Algorithm 5: feedback_criterion (Adjacency Matrix)

Function: feedback_criterion (Adjacency Matrix)

Input: Adjacency Matrix, $n \times n, n \in N$

Foreach $r_i \in$ register_set (R) do

  {

  $C5[r_i] = 0$

  Foreach $r'_i \in$ r_output (r_i) do

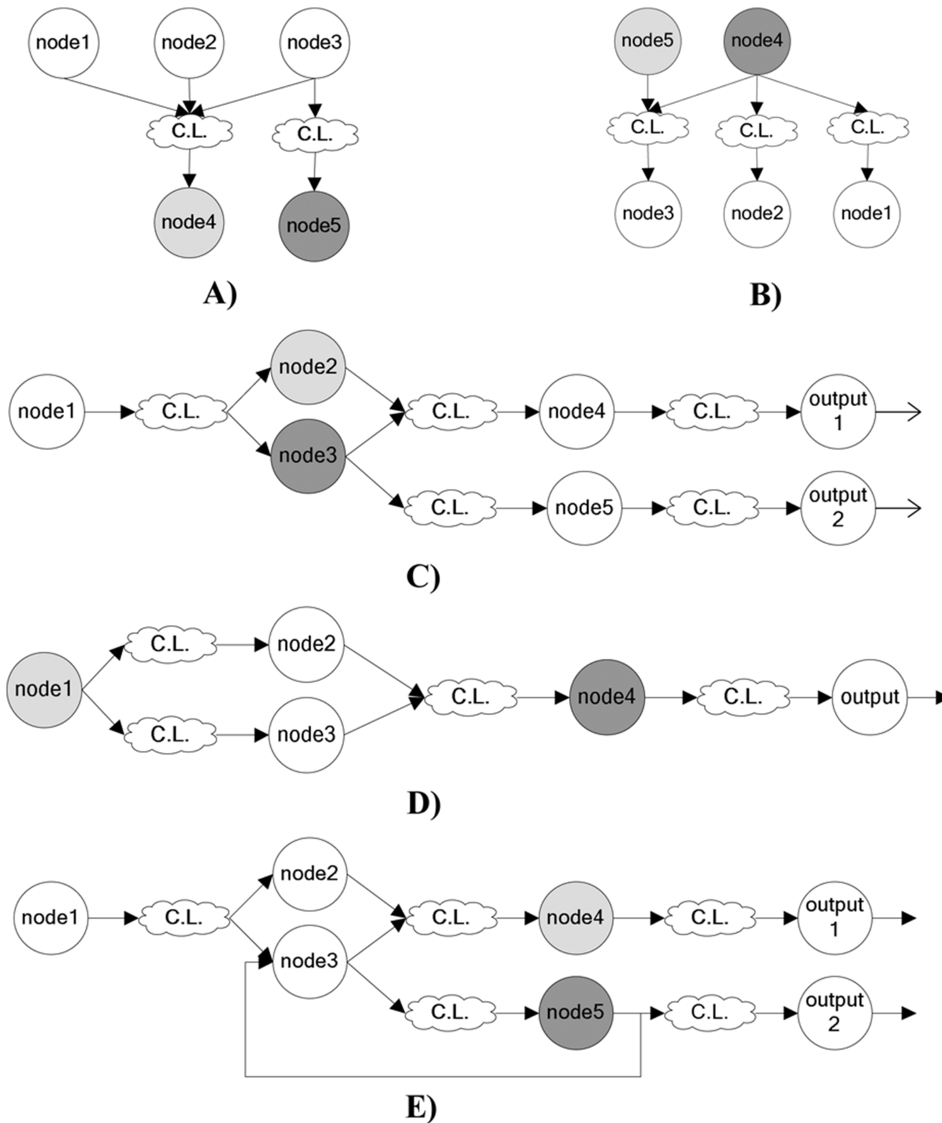

Fig. 7. Criteria Summary. Dark grey node: More sensitive FF, Light grey node: Less sensitive FF. (A) Input parameter. (B) Output parameter. (C) Influence on output paths.(D) Proximity to the output. (E) Feedback loop.

```
    {
        aux = dijkstra(Adjacency Matrix, r'_i, r_i)
        // Count the feedback paths
        If (aux < ∞) C5[r_i] ++
        }
}
    Return C5
```

In Fig. 7, a summary of the criteria which have been introduced in this section is shown. In each example, the expected behavior of the highlighted target flip-flops is estimated against the injection of SEUs through a color code. Dark grey nodes are expected to be more sensitive than light grey nodes according to the explained criteria. These criteria will be analyzed experimentally through fault injection in the following section, in order to validate the expected behaviors.

Finally, these algorithms have been implemented in order to obtain, in an automatic way, a criteria matrix, where each flip-flop is quantified for each of these criteria.

V. Experimental Analysis of the Criteria

In this section, the behavior of each criterion is verified experimentally, in order to guarantee the explained logic principles of the previous section. A set of specific circuits have been used to make the fault injection and prove the expected behavior. These circuits have been selected in order to measure the influence of each criterion independently (Fig. 7). Next, the obtained results achieved through fault injection are shown in Figs. 8 to 12. The results illustrate the number of error cycles observed at the outputs (Y axis) after having injected 1000 SEUs per register.

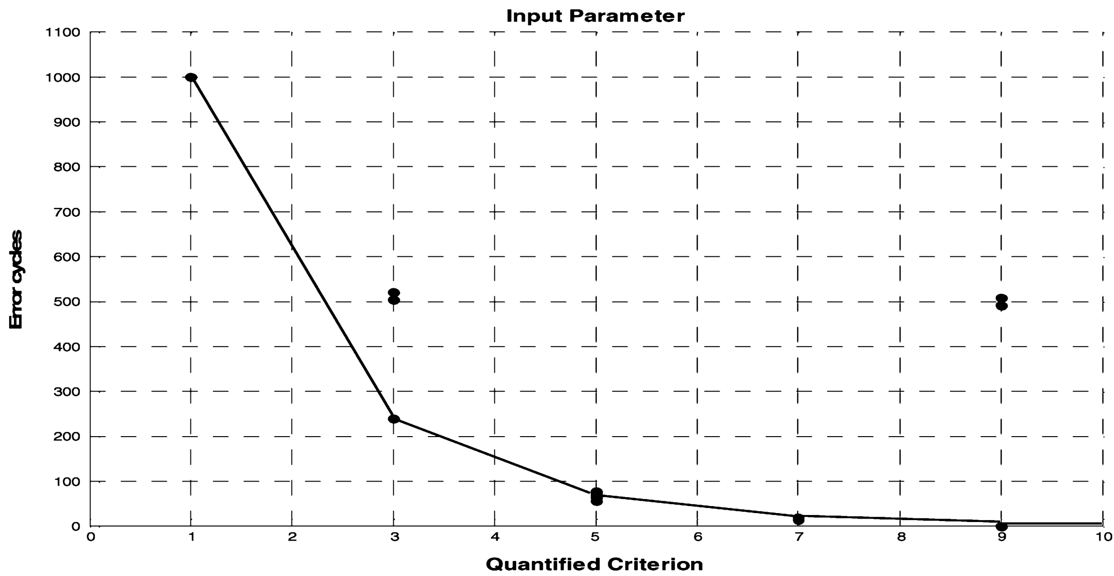

Fig. 8. Experimental analysis of FAN-IN.

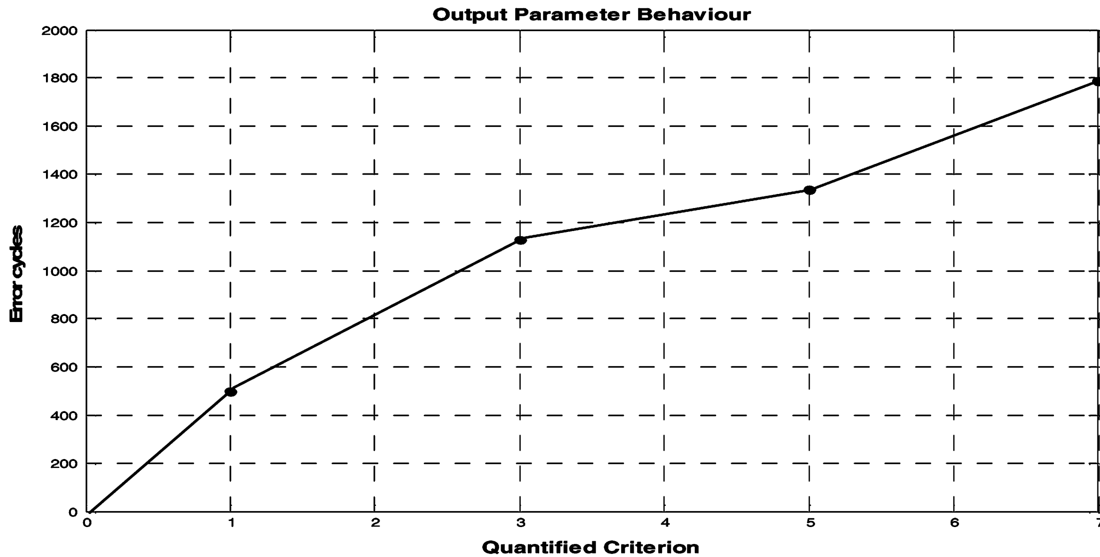

Fig. 9. Experimental analysis of FAN-OUT.

All these graphics represent the value of the associated criterion versus the measured error rate (wrong cycles/injected SEUs). With these results, it can be noticed that in all cases the behavior of the criteria corresponds with the expected one, explained in the previous section.

For the first criterion (fan-in), the trend is to decrease the error rate when the number of inputs that drive the flip-flop increases. In this case, it can be concluded that a flip-flop with a higher fan-in should not be tripled.

For criteria 2 and 3, the trend is to increase the error rate according to the increment of the number of paths/outputs respectively. In this case, it can be concluded that a flip-flop with a larger fan-out is susceptible to be tripled.

For criterion 4, the experimental results meet the expected behavior too. In this case, it can be observed that depending on the distance of the injected flip-flop to one primary output, the measured error rate increases. This is due to the filter capacity of the masking logic. In this case, it can be concluded that a flip-flop which is far from a primary output is not convenient to be tripled. Finally for the last criterion (feedback loop), the experimental results show a bigger vulnerability for those flip-flops that have larger feedback loops against others with smaller loops or the simple case without any loop. So it can be concluded that a flip-flop with feedback loops is more convenient to be tripled, depending on the size of the loop. As a result, taking into account the knowledge of the behavior against SEUs for all these common topological features, they can be used in an automatic way to propose an initial partition which is able to reduce the search space, thus improving the efficiency of the methodology.

VI. Quality Analysis of the Enhanced Technique

In order to verify the proposed enhancement, the same case study of Section III, is analyzed adding the criteria analysis. The

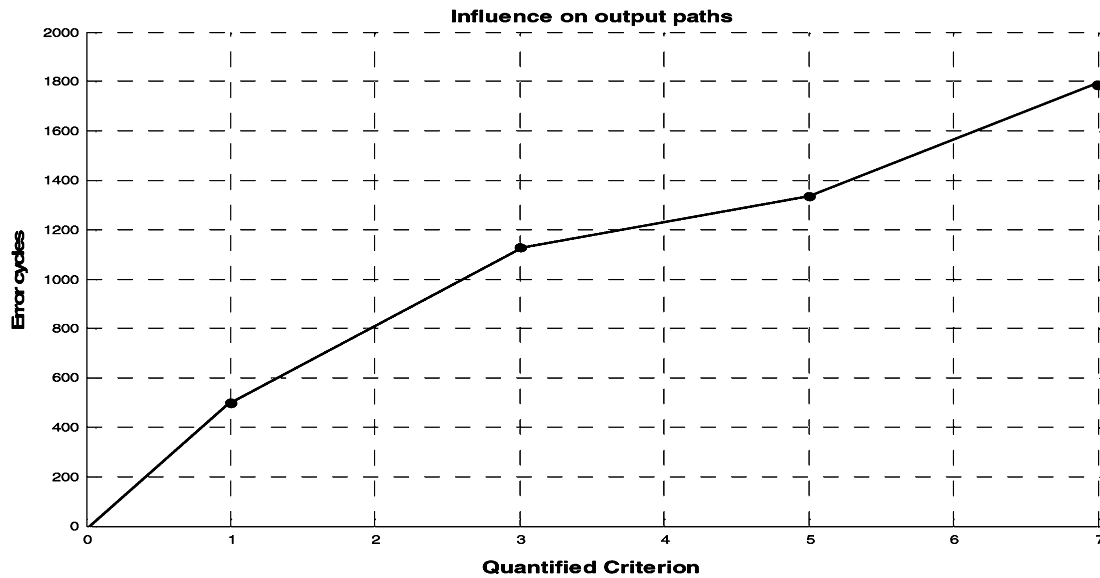

Fig. 10. Experimental analysis of the influence on output paths.

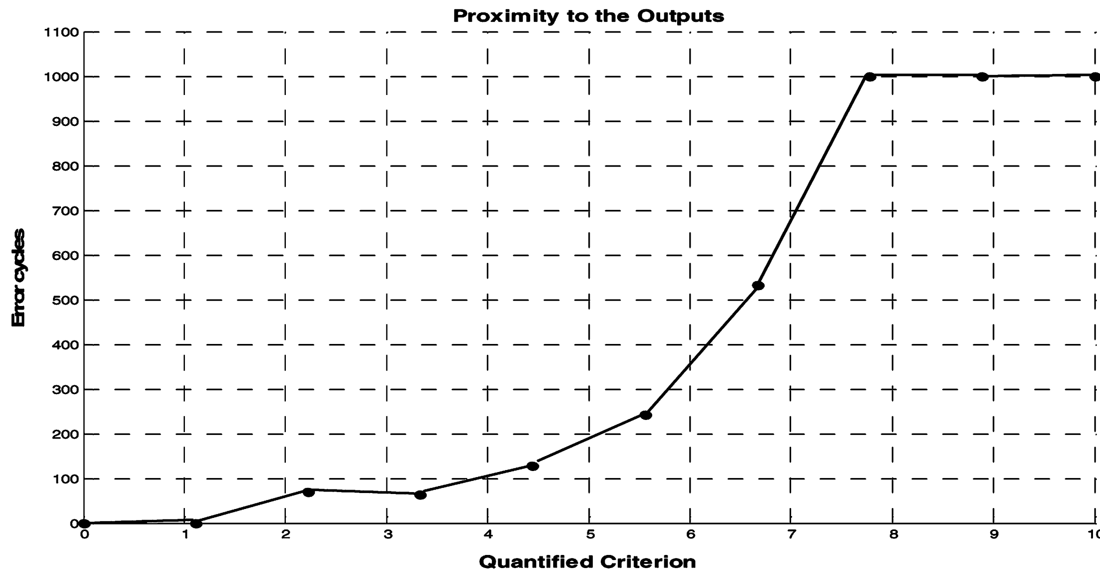

Fig. 11. Experimental analysis of the proximity to the output.

objective is to prove how the selection of a good initial partition, based on the mentioned criteria, can improve the efficiency of the optimization method. Now, for each reliability scenario proposed in Table I (0%, 0,5%, 1%) a similar test has been performed, where the only difference is the initial partition which is chosen according to the previous criteria set. To accomplish this, the algorithm set introduced in Section IV has been implemented in order to obtain a criteria matrix where the values of the criteria for each flip-flop are calculated. On this criteria matrix, once all flip-flops have been characterized, an algorithm has been applied in order to select those flip-flops which meet the next requirement. Any flip-flop is considered sensitive to SEUs if

— at least one of the criteria C2, C3 or C5 has the maximum value;
— the value of criterion C4 is high and has to affect at least a high-priority output;
— the value of criterion C1 is not high.

Applying these constraints, the following initial partition has been chosen by the automatic algorithm

$$\text{With TMR } \{\text{a_hold}, \text{indataready_delay}, \text{enb}, \text{enc}, \text{ena_delay}, \text{m_delay}, \text{out_delay}\}$$

$$\text{Without TMR } \{\text{a_delay}, \text{ena}, \text{b_hold}, \text{c_hold}\}.$$

Next, this initial partition will be justified following the previous criteria. For A_hold, Indataready_delay, Enb, and Enc, the reason is because they have the maximum fan-out value. In addition, Indataready_delay, Enb, and Enc affect the two primary outputs of the system, and therefore they are triplicated. For Ena_delay, M_delay and Out_delay, the proximity to the primary

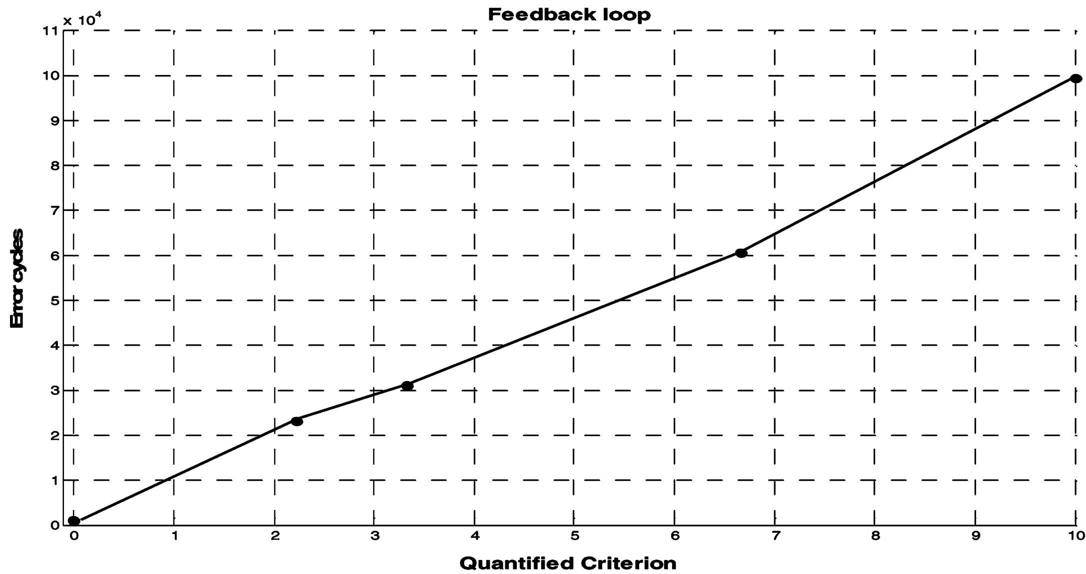

Fig. 12. Experimental analysis of the feedback loop.

TABLE II
COMPARISON OF PARTIAL SOLUTIONS EXPLORED WITH OR WITHOUT ENHANCEMENT

% Allowed wrong cycles	Detected wrong cycles		Explored partial solutions			
			With Enhancement	Without enhancement		
	Abs. value	% value	Topological choice	Random	None TMR	All TMR
0%	0	0%	**46**	58	103	12
0,5%	478	0,48%	**25**	28	73	39
1%	932	0,93%	**23**	28	71	46

output has been useful. Ena_delay is selected because its cost of triplication is lower than A_delay, which has the same value for criterion C4, but it is formed by more flip-flops. The rest of registers make up the no-TMR sub-graph. For example Ena has a higher fan-in so the first criterion (fan-in) makes it unnecessary to be tripled.

Finally, this initial partition has been used to perform the optimization process again on the case study. In Table II the experimental results with this new partition (29–25) are shown, in order to compare the use of the topological criteria versus the use of trivial initial partitions.

With these results, it can be concluded that this methodology helps the optimization engine to improve its efficiency, obtaining the same results in terms of cost as that in Table I (final partition). The only case where the technique does not improve the performance is the specific situation where 0% of wrong cycles are required. In this case, it is clear that starting with all registers in TMR would be faster. Nevertheless, if a certain degree of fault tolerance is allowed, which is the usual situation in real applications, using the topological criteria to find a good initial partitioning is the most convenient approach. For this example, the search space is rather small and therefore performance has been improved by a small margin. But for larger designs, this technique would be able to provide higher time savings by finding an initial partition closer to the optimal solution.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, a new methodology to insert selective TMR for SEU mitigation has been presented. The benefits of applying the automatic selective TMR methodology are clearly proved with the experimental results that have been obtained: the technique results in a lower circuit complexity than the traditional TMR approach, while meeting the specified reliability level. Besides, this methodology has been improved in terms of performance, adding an enhancement based on an innovative topological analysis of the target circuit. The benefit of this approach has been verified through fault injection experiments. This improvement results in a reduction of the number of explored partial solutions by the optimization engine, due to the calculation of an initial partition close enough to the optimal solution.

The next steps will focus on the study of these criteria in general purpose circuits and real space circuits like SpaceWire [27]. Also, a more sophisticated model that studies all the calculated criteria and automatically produces the set of conditions to initially map or not any given flip-flop in TMR will be produced. Finally the study of other optimization algorithms (simulated annealing, genetic, mimetic, adaptive), and alternative cost functions that can be adapted prioritizing the influence of each output will be carried out.

References

[1] Y. Moreau, S. Duzellier, and J. Gasiot, "Evaluation of the upset risk in CMOS SRAM through full three dimensional simulation," *IEEE Trans. Nucl. Sci.*, vol. 42, no. 6, pt. 1, pp. 1789–1796, Dec. 1995.

[2] F. Irom, F. F. Farmanesh, A. H. Johnston, G. M. Swift, and D. G. Millward, "Single-event upset in commercial silicon-on-insulator PowerPC microprocessors," *IEEE Trans. Nucl. Sci.*, vol. 49, no. 6, pt. 1, pp. 3148–3155, Dec. 2002.

[3] C. Carmichael, "Triple modular redundancy design techniques for virtex series FPGA," presented at the Application Notes 197, San Jose, CA, 2000, Xilinx.

[4] R. Hentschke, F. Marques, F. Lima, L. Carro, A. Susin, and R. Reis, "Analysing area an performance penalty of protecting different digital modules with hamming code and triple modular redundancy," presented at the Symp. Integrated Circuits and Systems Design, Proceedings, Sep. 2002.

[5] R. Velazco, D. Bessot, S. Duzellier, R. Ecoffet, and R. Koga, "Two CMOS memory cells suitable for the design of SEU-tolerant VLSI circuits," *IEEE Trans. Nucl. Sci.*, vol. 41, no. 6, pt. 1, pp. 2229–2234, Dec. 1994.

[6] L. Sterpone and M. Violante, "A new partial reconfiguration-based fault-injection system to evaluate SEU effects in SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pt. 2, pp. 965–970, Aug. 2007.

[7] F. Lima, S. Rezgui, E. Cota, L. Carro, M. Lubaszewski, R. Reis, and R. Velazco, "Designing a radiation hardened 8051-like micro-controller," in *Proc. Symp. Integrated Circuits and Systems Design*, Sep. 2000, pp. 255–260.

[8] P. K. Samudrala, J. Ramos, and S. Katkoori, "Selective triple modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 10, pp. 2957–2969, Oct. 2004.

[9] P. K. Samudrala, J. Ramos, and S. Katkoori, "Method and Apparatus for Creating Circuit Redundancy in Programmable Logic Devices," U.S. Patent Pub. No. WO/2004/077260, Oct. 2004.

[10] S. Krishnaswamy, S. M. Plaza, I. L. Markov, and J. P. Hayes, "Enhancing design robustness with realiability-aware resynthesis and logic simulation," in *Proc. Int. Conf. Computer Aided Design*, pp. 149–154.

[11] S. Krishnaswamy, I. L. Markov, and J. P. Hayes, "On the role of timing masking in reliable logic circuit design," presented at the Design Automation Conf., 2008.

[12] S. Almukhaizim and Y. Makris, "Soft error mitigation through selective addition of functionally redundant wires," *IEEE Trans. Rel.*, vol. 57, no. 1, pp. 23–31, Mar. 2008.

[13] B. Pratt, M. Caffrey, J. F. Carroll, P. Graham, K. Morgan, and M. Wirthlin, "Fine-grain SEU mitigation for FPGAs using partial TMR," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 4, pt. 1, pp. 2274–2280, Aug. 2008.

[14] F. Lima, G. Neuberger, R. Hentschke, L. Carro, and R. Reis, "Designing fault-tolerant techniques for SRAM-based FPGAs," *IEEE Des. Test Comput.*, vol. 21, no. 6, pp. 552–562, Nov.–Dec. 2004.

[15] P. Reyes, P. Reviriego, J. A. Maestro, and O. Ruano, "New protection techniques against SEUs for moving average filters in a radiation environment," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 957–964, Aug. 2007.

[16] O. Ruano, P. Reviriego, and J. A. Maestro, "A new EDAC technique against soft errors based on pulse detectors," presented at the IEEE Int. Symp. Industrial Electronics, Cambridge, U.K., Jun. 2008.

[17] D. Gonzalez, Single Event Upset Simulation Tool Functional Description ESA Rep. TEC-EDM/DCC-SST2, Jul. 2004.

[18] O. Ruano, J. A. Maestro, P. Reyes, and P. Reviriego, "A simulation platform for the study of soft errors on signal processing circuits through software fault injection," in *Proc. IEEE Int. Symp. Industrial Electronics*, 2007, pp. 3316–3321.

[19] J. C. Baraza, J. Gracia, S. Blanc, D. Gil, and P. J. Gil, "Enhancement of fault injection techniques based on the modification of VHDL code," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 6, pp. 693–706, Jun. 2008.

[20] C. Fiduccia and R. Mattheyses, "A linear time heuristic for improving network partitions," in *Proc. 19th IEEE Design Automation Conf.*, 1982, pp. 175–181.

[21] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. New York: Springer-Verlag, 2000, 978-0-7923-7991-1.

[22] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms," *IEEE Trans. Comput.*, vol. C-32, no. 12, pp. 1137–1144, Dec. 1983.

[23] D. J. Smith, *HDL Chip Design. A Practical Guide for Designing, Sythesizing and Simulating ASICs and FPGAs Using VHDL or Verilog* . Madison, AL: Doone, 2001, ISBN 0-9651934-3-8.

[24] J. L. Gross and J. Yellen, *Grahp Theory and Its Applications*. Boca Raton, FL: Chapman & Hall/CRC, ISBN: 158488505X.

[25] E. Dijkstra, "A note on the problems in connection with graphs," *Numer. Math.*, Oct. 1959.

[26] W. Stallings, *Data & Computer Communications*. Upper Saddle River, NJ: Pearson Prentice-Hall, 2007, ISBN-10: 0132433109.

[27] SpaceWire. [Online]. Available: http://spacewire.esa.int/content/Home/HomeIntro.php