Addressing Single-Event-Multiple-Transient Faults in Asynchronous RH-Click Controllers

Felipe A. Kuentzer*, Christos Georgakidis‡, Christos Sotiriou‡, and Milos Krstic*†
* Leibniz-Institut für Innovative Mikroelektronik (IHP), 15236 Frankfurt, Germany
† University of Potsdam, 14469 Potsdam, Germany
{kuentzer, krstic}@ihp-microelectronics.com
‡ Department of Electrical and Computer Engineering, University of Thessaly, 38221 Volos, Greece
{cgeorgakidis, chsotiriou}@e-ce.uth.gr

2023 36th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI) | 979-8-3503-1834-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/SBCCI60457.2023.10261666

***Abstract*—Technology shrinking has allowed major improvements through the last decades, from the increased performance and design complexity while maintaining circuit area to more reliable and power-efficient Integrated Circuits (ICs). However, with smaller transistors, especially in harsh environments, modern chips are becoming more susceptible to Single-Event-Multiple-Transients (SEMTs) in the combinational logic due to radiation effects. The combination of asynchronous design with Radiation Hardened by Design (RHBD) has shown great potential to increase robustness to soft errors arising from Single-Event-Transients (SETs), but detailed SEMT analyses are missing and of great concern for the future that lies ahead of modern VLSI systems. In this paper, the asynchronous Radiation Hardened (RH) Click controller's robustness to SEMT is accessed, and a post-placement approach is proposed to group and space critical cells apart. Fault simulation experiments demonstrate that the proposed spacing strategy is effective in mitigating SEMTs.**

***Index Terms*—Asynchronous Design, SEMT, Click Controller, Placement, Soft Errors, Radiation Hardening, Reliability.**

I. Introduction

THE INCREASING sensitivity of modern CMOS technologies to soft errors can arise from single factors or a combination of several. The technology down-scaling trend is one of the fundamental factors affecting sensitivity. Smaller transistors present reduced node capacitance and make them more susceptible to the effects of radiation. Combining the process, voltage, and temperature variations, circuit aging, and the potential applications targeting harsh environments forms the perfect scenario for soft errors to emerge.

In reliability-critical applications, such as those in space environments, radiation-induced effects are one of the main challenges to consider when designing digital Integrated Circuits (ICs). Space applications are susceptible to ionizing particles (*i.e.* protons, neutrons, heavy-ions, *etc.*) strikes that can lead to Total Ionising Dose (TID) and Single-Event Effects (SEEs). TID effects may degrade operating parameters such as leakage and threshold voltage [1] [2], which impacts the device performance. Methods to mitigate TID effects are usually implemented at the technology or layout level, *e.g.* the use of Enclosed Layout Transistors (ELT) [3]. The typical SEEs include Single-Event-Transients (SETs), *i.e.* short voltage pulse in the combinational logic, and Single-Event-Upsets (SEUs), *i.e.* when a SET is latched by a memory element.

Several approaches have been proposed in the literature to mitigate SEEs, from the traditional Triple Modular Redundancy (TMR) approach to other alternatives which include gate sizing [4] [5], glitch filtering [6] [7], logical masking [8] [9], and built-in soft error-resilient architectures, such as Full Error Detection and Correction (FEDC) [10] and Soft Error Resilient Asynchronous Design (SERAD) [11]. Nevertheless, considering only the simplified and unrealistic model of single events is no longer sufficient.

Besides the increased sensitivity to high-energy particles, transistor shrinking has reduced the distance between cells junctions and thus increasing the chance of Single-Event-Multiple-Transients (SEMT). SEMT is the result of a single particle strike affecting several adjacent cells (*i.e.*, due to collected charge sharing) and producing multiple SETs. The problem of SEMT has been addressed first in memories [12] [13], and in the past decade, these effects have gained more attention in logic circuits [14] [15] [16].

Combining Radiation Hardened by Design (RHBD) with asynchronous design, has recently shown to be a promising solution to improve robustness to soft errors in modern VLSI systems while reducing the area, power, and/or performance overheads typically observed in traditional RHBD approaches. Among these promising solutions is the Asynchronous Full Error Detection and Correction (AFEDC) [17]. The AFEDC is an asynchronous soft error-resilient architecture with Radiation Hardened (RH) Click controllers [18] in the control path. The RH-Click controller employs space redundancy to mitigate SETs and SEUs, but until now, its tolerance to SEMT remains untested.

This paper evaluates the RH-Click controller's robustness to SEMT and proposes combining RHBD with a post-placement strategy for spacing the critical cells to improve its reliability by reducing the probability of multiple transients (MTs) affecting the controller's behavior. The spacing constraints are defined based on a spacing strategy which consists of two incremental steps. The first step groups cells, taking into account the controller's functionality and existing redundancies. The second step selects critical cells based on fault simulations of multiple transient (MT) combinations and defines additional groups for spacing. To the best of our knowledge, this is the first work to address SEMT in the context of asynchronous circuit design.

The rest of the paper is organized as follows: Section II highlights the SEMT state of the art and presents the

motivation behind the present work; Section III presents an overview of the RH-Click controller design, which serves as a case study in this work, as well as the SEMT vulnerability analysis; Section IV introduces the proposed spacing approach to improve the RH-Click robustness to SEMTs; Section V briefly describes the design flow and shows the proposed methodology applied to the case study along with the fault experiments results that demonstrate the proposed strategy effectiveness to mitigate SEMT.Finally, Section VI concludes this work.

II. Related Works and Motivation

The AFEDC [17] is a soft error-resilient architecture that can tolerate SETs and timing violations [19] of unbounded duration and relies on an Error Detection Circuit (EDC) designed with the Full Duplication and Comparison (FDC) method to protect the datapath. It also addresses the metastability issues that in the past were neglected by similar synchronous solutions [20] and presents lower area and power overheads when compared to equivalent synchronous and traditional TMR approaches. The AFEDC custom design flow starts with the synchronous description of a circuit. The circuit is synthesized using commercial EDA tools and converted to the asynchronous design, where the global clock signal is removed and replaced by RH-Click controllers. These controllers are RHBD following the methodology described in [18]. Despite the advantages of the AFEDC architecture, ensuring only robustness to SETs and not considering SEMTs may limit its range of applications. In particular, SEMTs in the asynchronous control path are of great concern, as further demonstrated in Section V, especially when unexpected transitions can put the controller into a deadlock state from which it cannot recover.

Not many works in the literature address SEMT in the asynchronous domain, especially when it comes to mitigation. The work [21] is one of the few available evaluating the tolerance of an asynchronous architecture against multiple faults. Results indicate the circuit's vulnerability to SEMT, however, not all SEMTs may be tolerated, and no mitigation approach is proposed to prevent these faults from affecting the circuit behavior. Different aspects are considered when addressing SEMT in the combinational logic, the modeling of Transient Fault (TF) propagation [14], SER estimation [22], measurements [23], and mitigation of effects [24]. Fault modeling and SER estimation gained a lot of attention [15] [25] [26] [27] [28], but despite being crucial to determine the vulnerability of the ICs, only a few approaches draw attention to SEMT mitigation techniques [16] [24].

Spacing cells is a promising alternative for SEMT mitigation, as it allows relatively easy integration with commercial P&R tools in custom design flows [29]–[31]. The challenge remains in properly selecting the cells for spacing in order to minimize the area and timing overheads associated with the achieved robustness. In [16], two post-placement strategies to mitigate SEMTs are presented, the All-to-All, in which all cells are evenly spaced among each other, and the TMR-based, where only critical cells are tripled and spaced among the redundant cells. Fault simulations are conducted, taking into account particle strikes with different energies at random circuit parts, generation of SEMT, and its propagation through the circuit logic considering different masking mechanisms. While providing accurate SER estimations for large circuits, it should be noted that for small circuits, very small sections may be created, thus the extracted SER data may be misleading [16], and therefore not suited for asynchronous controllers.

Considering the context above, this paper assesses the vulnerability of the AFEDC's RH-Click controller against SEMTs, and presents a strategy that combines RHBD with a post-placement strategy for selecting and spacing critical groups of cells so as to mitigate SEMT effects. As will be further demonstrated, strategies such as the All-to-All spacing and TMR spacing presented in [16] would add unnecessary space overhead or unnecessary area redundancy, respectively. Moreover, the combinational logic in the RH-Click design is very small. Thus, the SER data extraction method presented in [16] may not be suitable for the target circuit. Instead, a faults' simulation environment which integrates the Cadence Incisive Functional Safety Simulator (IFSS), and custom scripts are used to extract the error results.

III. RH-Click Controller

The RH-Click controller is implemented using the asynchronous bundled-data design style based on the Click [32] template. Bundled-data designs use standard Boolean encoding to represent information and separate request and acknowledge wires are bundled with data signals to provide synchronization. The communication is usually implemented with 2-phase or 4-phase handshake protocols [33]. The 2-phase and 4-phase terms refer to the number of signal transitions required to complete the communication cycle. For instance, in a 2-phase implementation, the communication starts with the rising edge of the request signal and ends with the rising edge of the acknowledge signal, and the next communication cycle starts with the falling edge of the request signal and ends with the falling edge of the acknowledge signal.

This work's case study, *i.e.* the non-RH-Click controller, designed for the AFEDC architecture, is shown in Fig. 1. It implements a 2-phase bundled-data channel for communication and also controls the AFEDC datapath error sampling and recovery. The timing diagram of Fig. 2 illustrates the controller's general behavior for one error-free communication

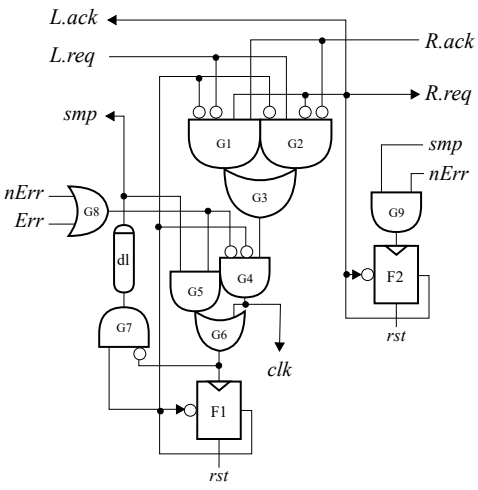

Fig. 1: AFEDC non-RH-Click controller implementation, adapted from [17].

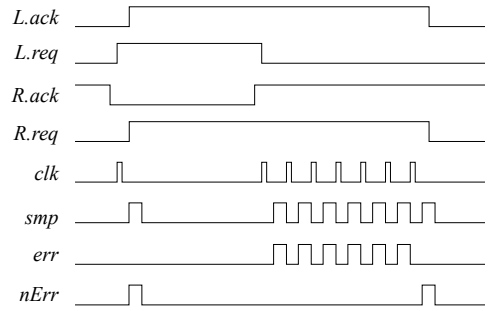

Fig. 2: AFEDC timing diagram, adapted from [17].

and one possible error detection and recovery process. Note that request (*R.req*) and acknowledgment (*L.ack*) transitions are only generated if no error (*nErr)* is flagged in the AFEDC datapath. If an error (*err*) is sampled (*smp*), the controller enters into a re-sampling loop and remains in this state until the error in the datapath disappears. This mechanism guarantees that TFs of unbounded duration are tolerated. For more information on the AFEDC architecture, a detailed description is available in [17].

Two flip-flops (FFs), *F1* and *F2*, represent the controller's four internal states. The initial state after reset is $(0, 0)$. This state enables a rising edge of *L.req* to propagate through gates *G2* and *G3*. Next, *G4* transitions to 1, asserting *clk* high, and through *G6* the *F1* clock input is asserted. The *F1* input port is connected to its inverted output, so the controller transitions to state $(1, 0)$. This makes the *G4* transition to 0 and the *F1* input clock falls as well as the *clk* signal. The *F1* output prevents request and acknowledge signals to propagate through *G1* and *G3*, and also propagates through the *G7* and the delay line *dl*, causing *smp* to rise, thereby sampling the error signals *Err* and *nErr*. If the *nErr* is subsequently received, indicating that no error was detected, *G9* transitions to 1, asserting the *F2* clock input. In parallel, *nErr* propagates through *G8*, *G5* and *G6*, asserting the *F1* clock input. At this point, the controller transitions to the state $(0, 1)$. If an error was detected, the propagation of *Err* down *G8* would have caused a transition back to the state $(0, 0)$. From state $(0, 1)$, the *F2* output port now enables the path through *G1*, and a falling edge of *L.req* starts a new communication instead. In this new communication cycle, the controller will first transition to state $(1, 1)$ and, in the end, return to the initial state $(0, 0)$.

A. Hardening Method

As previously mentioned, the RH-Click controller is implemented according to the generalized RHBD method presented in [18]. The baseline for hardening starts with the previously described non-RH-Click shown in Fig. 1. The method relies on spatial redundancy and the use of Guard Gates (GGs) [6] as voters, which output, similar to a C-element, only changes when both inputs have the same value. Fig. 3 shows a generalized RH-Click controller. The combinational logic and the sequential elements (FFs) are fully duplicated. After duplication, all equivalent FF outputs and clock inputs are cross-coupled with GGs, and GGs are added for "voting" equivalent control signals. The GGs prevent SETs from propagating to sensitive parts of the controller, such as from the controller's combinational logic to the FFs, which would cause an undesired state transition (soft error).

B. SEMT Analysis

Before proposing any SEMT mitigation technique, it is essential to analyze the impact of multiple TFs in the current RH-Click design, as it can provide valuable information on the relevant faults and what are the potential critical gates. Considering that in the RH-Click design, there is a copy of each element shown in Fig. 1, it would be possible to state that, as long as a particle strike does not affect the same copies, the controller would tolerate MTs. Moreover, it could also be stated that as long as one of the combinational block copies remains unaffected, the other could present any number of concurrent TFs. In this case, one could argue that simply spacing the two combinational copies and register copies, thus having two spatially mutually-separated logic clouds, would be enough to solve the SEMT problems in the RH-Click design, but these statements are only partially correct.

Assume the combination block *C1* and its copy *C2*. If the first statement is true, then *G1* in *C1* and *G2* in *C2* can be affected by the same particle strike. However, they belong to the same logical path to *F1*, and even though the TFs originate from different gate copies, the fault would propagate to both GGs inputs, allowing the RH-Click to have an invalid state transition. One possible solution could be, in addition to spacing *G1* copies, also to space *G1* copies from the *G2* copies, but this is not an optimal solution if *G1* and *G2* of *C1* are simultaneously affected but the copies of *C2* are not, then the controller behavior would not be affected. The ideal solution, in this case, is to only space *G1* and *G2* that belong to a different combinational block and allow them to be placed closely if they belong to the same block.

The previous analysis supports our second statement that simply spacing two combinational copies is enough. Nevertheless, some combinations of MTs in sync with specific timing conditions can expose vulnerability, even if MTs are contained within the same combinational copy. One of these cases is when *G8* and *G5* of *C1* or *C2* are affected at the same time that a communication cycle begins. Assuming that the communication cycle begins with the *L.req* rising edge. Both *C1* and *C2* will propagate this transition through their *G2*. If *G8* and *G5* of *C1* are affected, the transient fault coming from *G8* masks the *G2* transition through *G4*, and prevents *C1* to assert its *clk* high. On the other hand, *C2* correctly asserts the *clk* high, but since only the *C2 clk* is high, the transition is not propagated through the GG. The major problem arises from gate *G5*. In this scenario, the faulty transition coming from *G5* is propagated through *G6*, and combined with the correct behavior of *C2*, ends up asserting the clock input of both *F1* copies. The controller then transitions to the state $(1, 0)$, and the communication cycle completes successfully even though no *clk* was propagated to the datapath. No errors are flagged since no new data was stored for comparison in the datapath registers.

Considering these vulnerabilities, it is clear that the RH-Click controller requires more sophisticated hardening to tolerate SEMT. In particular, a strategy to satisfy spacing constraints based on analyzing MT fault combinations and identifying critical cells.

IV. Proposed Spacing Strategy

The proposed approach to address SEMT in asynchronous RH-Click combines the RHBD described in [18] with a post-

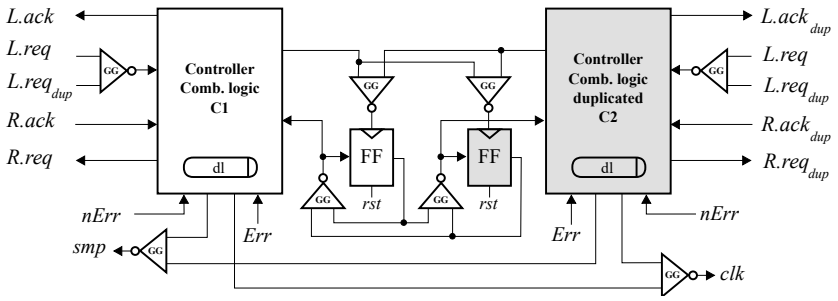

Fig. 3: Generalized RH-Click controller, adapted from [18].

placement strategy for spacing critical cells. As demonstrated in Section III-B, some groups of cells in the RH-Click controller can be placed close to each other, thus minimizing area and timing overheads associated with the spacing strategy. Moreover, the controller's timing behavior is crucial for extracting the critical MT combinations and tracing the related critical cells. Therefore, the approach is divided into two incremental steps. The first step relates to the grouping part of the strategy, and the second relates to the tracing of critical cells.

A. Grouping Step

The grouping phase occurs during the design description of the non-RH-Click, where cells belonging to a given group are separated into hierarchical logic blocks that are preserved during the RH-Click synthesis. The division of cells between groups is defined by the controller's state transitions. Whenever a gate or a set of gates controls a state transition, this gate or the set of gates becomes a group. For instance, considering the initial state of $F1$ and $F2$ (Fig. 1) to be $(0,0)$, the path from *G1* or *G2* through *G3*, *G4*, and *G6* controls the transition $(0,0) \rightarrow (1,0)$, while the path through *G8*,*G5*, and *G6* control's the transition $(1,0) \rightarrow (0,0)$. If a gate is present in more than one path, it is allocated to a separate group, such as *G6*. In case a single gate controls a FF transition, then the gate alone forms a group. This is the case of *G9*.

Once all groups are defined, the RH-Click controller is synthesized, and the spacing constraints are applied during post-layout placement. After the grouping step, the only spacing rule is that copies of each group must be spaced at a certain distance. Even though sequential elements are not addressed in this work, $F1$ and $F2$ could also be defined as separate single element groups and constrained to be spaced out from their copies.

B. Tracing Step

The tracing of critical cells consists of simulating the injection of MT fault combinations and identifying the gates with higher incidence of errors associated with them.The faults simulations performed in this step are not injected at random locations. Instead, an exhaustive scenario is assumed, and all possible MT combinations are simulated. The random parameters are the fault pulse width and injection time of the faults.

Fault simulations are performed with the post-placement netlist and annotated gate and wire delays. This type of simulation helps identify the MT combinations that generated errors related to the controller's timing behavior, such as the one involving *G8* and *G5* (Fig. 1), and that could only be detected in this step. When fault the simulations are complete, the results are automatically evaluated. The list of fault combinations which lead to errors is exported along with the number of times each gate appeared in these combinations, and this represent the critical gates. At this point, the designer determines which gates or groups of gates are relevant for spacing and from which other gates these should be spaced. For instance, considering the *G8* and *G5* case, these gates are not allowed to be placed close to each other, so the group defined in the previous step cannot be sustained, as cells belonging to the same group are allowed to be placed near each other. One possible solution is to create two separate groups, $\{G5\}$ and $\{G8\}$, and specify that these groups should be spaced from their copies and from each other inside the same combinational copy.

Next, we used the fault simulation flow which combines the IFSS tool with custom scripts to generate the fault campaigns. These scripts handle the random pulse width for each fault injection and generate all possible MT combinations based on the fault list file (all gates' output path), which is generated by the IFSS tool. The total number of fault simulations of each campaign depends on the set of possible combinations that can be formed according to the specified MT size. The IFSS runs the golden simulation, handles fault injections at random times, and generates the simulation logs and reports. The generated logs and reports are then evaluated to extract critical gates.

V. Spacing Strategy Evaluation

The RH-Click controller robustness to SEMT and the proposed mitigation approaches were verified and evaluated using a linear 5-stage pipeline version of a 32-bit multiplier and the AFEDC controller's design as the case study. The asynchronous multiplier is generated through an asynchronous design flow [17]. The flow consists of Synopsys Design Compiler and a set of custom TCL and Shell scripts that are used to synthesize the circuit targeting a 130nm technology process. For faults simulations, the Cadence Incisive Functional Safety Simulator (IFSS) tool is also combined with custom scripts to generate the fault campaigns. These scripts handle the random pulse width for each fault injection and generate all possible MT combinations based on the fault list file (all gates' output path), which is a file generated by the IFSS tool. The total number of fault simulations of each campaign depends on the set of possible combinations that can be formed according to the specified MT size. The IFSS runs the golden simulation, handles fault injections at random times, and generates the

simulation logs and reports. The generated logs and reports are then evaluated to extract critical gates.

It is important to mention that in this evaluation, the tool required to satisfy the spacing constraints is not addressed. Moreover, no major effect would be observed when running fault simulations with different spacing constraints since the positions of the cells are not available in the netlist. Therefore, to assess the SEMTs mitigation impact, the simulation flow analyzes the MT combinations that would be injected, and it marks those combinations that are forbidden according to the spacing strategy (IV), thus assuming the spacing are satisfied. For instance, if two gates can be placed close to each other, the MT combination where these two gates are affected at the same time is valid, and it is added to the fault campaign, while if these gates are spaced out, the flow assumes they would never be affected simultaneously by a single particle strike, so the MT combination is marked as invalid and removed from the fault campaign.

The subsequent analyses are focused on the controllers, and the datapath serves as an observation point for the side effects of the faults injected in the control path. Next, the methodology proposed in Sec. IV is applied in the AFEDC controller described in Sec. III.

A. Methodology Application

***Step 1 - Grouping Step**:* The first grouping step is based on the controller's state transition. Considering the design of Fig. 1, the gates involved in a valid transition of *F1* from 0 to 1 are *G1*, *G2*, *G3*, *G4*, and *G6*, while gates *G8*, *G5*, and *G6* control the transition back from 1 to 0. Based on this, three groups are formed, $\{G1, G2, G3, G4\}$, $\{G8, G5\}$, and $\{G6\}$ alone since it would intersect both groups. Note that there is a path from *G8* through *G4*. However, this path does not generate a valid state transition of *F1*. *G4* is added in this design actually to block any transition from *G3*, so it is not separated. The same applies to *G5* since it blocks transitions coming from *G8*. The fourth group is composed of *G7* and all the cells that are used to implement the delay line *dl*. This group causes an external state transition when *smp* is asserted, and the error signals are registered in the datapath. The last group $\{G9\}$ alone, as it is the only gate controlling state transitions of *F2*. At this point, the first step of the proposed spacing strategy is concluded, and the formed groups are constrained to be spaced from their copies. The configuration of these groups and spacings are evaluated in the $T2$ experiment presented in the following subsection and is the initial configuration for the next step.

***Step 2 - Tracing Step**:* In the tracing step, MT fault simulations are used to extract the critical gates and identify cases where simply spacing group copies is not sufficient. The tracing analysis for the AFEDC RH-Click design revealed that all the MT combinations reporting errors had either *G8* or *G6* in them. As already described in Section III-B, *G8* and *G5* is one fault combination that requires special timing conditions to generate an error. Regarding *G6*, for instance, looking at Fig. 1, the combination of *G6* and *G3* should not be allowed, as a transient generated at *G3* propagating through *G4* could reach *G6* of one of the combinational copies (*e.g. C1*) at the same time that *G6* from the other copy (*e.g. C2*), thus propagating to the GG's input and causing an invalid state transition. These are only two of several MT combinations that report errors, and especially when increasing the number of concurrent TFs injected, the number of combinations with errors increases significantly. As a result of the tracing, $\{G8\}$ alone forms a separate group, and *G5* is integrated into group $\{G5, G7, dl\}$. *G5* could be left as a single group element, but this could also prevent it from being placed close to other cells. In addition, *G6* and *G8* are especially constrained to be spaced apart not only from their copies, but from all other gates which compose the controller design. The improvements achieved with this step are demonstrated next in the $T3$ experiment.

B. SEMT Fault Experiments

Three separate experiments for two MT values were conducted to assess the effectiveness of the spacing strategy. The first experiment ($T1$) is the baseline, where no spacing is applied, and all MT combinations are allowed. The second experiment ($T2$) includes only the groups and the spacing constraints formed in **Step 1** (Section V-A), while the third experiment ($T3$) is the result of tracing and spacing critical cells performed in **Step 2** (Section V-A). Even though there are five controllers, one for each pipeline stage, only the combinational logic of the controller in the center stage is the target of fault injections, and the errors reported are either incorrect values computed by the multiplier or cases where the circuit halted due to transients causing an invalid state transition.

Table I shows the fault simulation results considering 2 and 3 MT, and the reasoning for selecting this number of MTs is not related to limitations in the proposed mitigation strategy but rather due to the increased simulation time required to simulate all possible combinations with larger MTs. One possible MT combination is injected in each fault simulation, and the duration of the transients is randomly generated in a range from 100ps to 2000ps. According to **Step 1**, experiment $T2$ has the following groups, $\{G1, G2, G3, G4\}$, $\{G5, G8\}$, $\{G7, dl\}$, $\{G6\}$, and $\{G9\}$, and assume the spacing of these groups from their copies. The grouping approach reduced the percentage of errors, but a few errors are still reported. According to **Step 2**, the groups of $T3$ are $\{G1, G2, G3, G4\}$, $\{G5, G7, dl\}$, $\{G6\}$, $\{G8\}$, and $\{G9\}$, and in addition to spacing these groups from their copies, $\{G6\}$ and $\{G8\}$ are spaced from all other individual gates in the controller. $T3$ results show that with this spacing strategy, no MT combination generates an error.

TABLE I: MT fault simulation experiments for the RH-Click.

MT	Experiment	Simulations	Errors	%
2	T1	1891	29	1.53%
	T2	1376	3	0.22%
	T3	1032	0	0.00%
3	T1	37820	1726	4.56%
	T2	15388	135	0.88%
	T3	8644	0	0.00%

VI. Conclusions

This paper assessed the robustness of the RH-Click controllers to SEMT and proposed a post-placement approach to space cells and reduce the probability of SEMTs affecting the

controller's behavior. The spacing strategy proposed consisted of two incremental steps, the first for grouping cells based on the controller's state transition, which allows some cells to be placed close to each other and avoid unnecessary spacing overheads, and the second step finds the critical cells-based MT fault simulations. The evaluation demonstrated the effectiveness of the proposed strategy to prevent MTs from affecting the RH-Click behavior. The next step is to satisfy the spacing constraints following the proposed strategy and assess the area, power, and delay overheads associated with this approach in the circuit layout.

ACKNOWLEDGMENT

This work has been funded by the DFG-project ENROL (KR 4346/2-1) and the MORAL project (EU Horizon 2020 - GA 870365). The research was also elaborated in the operated framework of the Center of Research Innovation and Excellence of University of Thessaly (Invitation to submit applications for the grant of scholarships to doctoral candidates of University of Thessaly) and partially funded by the Special Account for Research Grants of University of Thessaly.

REFERENCES

[1] D. M. Fleetwood, "Total Ionizing Dose Effects in MOS and Low-Dose-Rate-Sensitive Linear-Bipolar Devices," *IEEE Transactions on Nuclear Science*, vol. 60, no. 3, pp. 1706–1730, 2013.

[2] M. P. King, X. Wu, M. Eller, S. Samavedam, M. R. Shaneyfelt, A. I. Silva, B. L. Draper, W. C. Rice, T. L. Meisenheimer, J. A. Felix, E. X. Zhang, T. D. Haeffner, D. R. Ball, K. J. Shetler, M. L. Alles, J. S. Kauppila, and L. W. Massengill, "Analysis of TID Process, Geometry, and Bias Condition Dependence in 14-nm FinFETs and Implications for RF and SRAM Performance," *IEEE Transactions on Nuclear Science*, vol. 64, no. 1, pp. 285–292, 2017.

[3] W. Snoeys, T. Gutierrez, and G. Anelli, "A new NMOS layout structure for radiation tolerance," in *2001 IEEE Nuclear Science Symposium Conference Record (Cat. No.01CH37310)*, vol. 2, 2001, pp. 822–826 vol.2.

[4] Quming Zhou and K. Mohanram, "Gate sizing to radiation harden combinational logic," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 1, pp. 155–166, Jan 2006.

[5] M. A. Sabet, B. Ghavami, and M. Raji, "A Scalable Solution to Soft Error Tolerant Circuit Design Using Partitioning-Based Gate Sizing," *IEEE Transactions on Reliability*, vol. 66, no. 1, pp. 245–256, Mar 2017.

[6] A. Balasubramanian, B. L. Bhuva, J. D. Black, and L. W. Massengill, "RHBD techniques for mitigating effects of single-event hits using guard-gates," *IEEE Transactions on Nuclear Science*, vol. 52, no. 6, pp. 2531–2535, Dec 2005.

[7] P. Mongkolkachit and B. Bhuva, "Design technique for mitigation of alpha-particle-induced single-event transients in combinational logic," *IEEE Transactions on Device and Materials Reliability*, vol. 3, no. 3, pp. 89–92, Sep 2003.

[8] A. H. El-Maleh and K. A. K. Daud, "Simulation-Based Method for Synthesizing Soft Error Tolerant Combinational Circuits," *IEEE Transactions on Reliability*, vol. 64, no. 3, pp. 935–948, 2015.

[9] K.-C. Wu and D. Marculescu, "A Low-Cost, Systematic Methodology for Soft Error Robustness of Logic Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 2, pp. 367–379, 2013.

[10] M. Krstic, S. Weidling, V. Petrovic, and E. S. Sogomonyan, "Enhanced architectures for soft error detection and correction in combinational and sequential circuits," *Microelectronics Reliability*, vol. 56, pp. 212–220, 2016.

[11] S. A. Aketi, S. Gupta, H. Cheng, J. Mekie, and P. A. Beerel, "SERAD: Soft Error Resilient Asynchronous Design Using a Bundled Data Protocol," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 5, pp. 1667–1677, 2020.

[12] R. C. Martin, N. M. Ghoniem, Y. Song, and J. S. Cable, "The Size Effect of Ion Charge Tracks on Single Event Multiple-Bit Upset," *IEEE Transactions on Nuclear Science*, vol. 34, no. 6, pp. 1305–1309, 1987.

[13] T. Merelle, F. Saigne, B. Sagnes, G. Gasiot, P. Roche, T. Carriere, M.-C. Palau, F. Wrobel, and J.-M. Palau, "Monte-Carlo simulations to quantify neutron-induced multiple bit upsets in advanced SRAMs," *IEEE Transactions on Nuclear Science*, vol. 52, no. 5, pp. 1538–1544, 2005.

[14] N. Miskov-Zivanov and D. Marculescu, "Multiple Transient Faults in Combinational and Sequential Circuits: A Systematic Approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 10, pp. 1614–1627, 2010.

[15] B. T. Kiddie and W. H. Robinson, "Alternative Standard Cell Placement Strategies for Single-Event Multiple-Transient Mitigation," in *2014 IEEE Computer Society Annual Symposium on VLSI*, 2014, pp. 589–594.

[16] C. Georgakidis, G. I. Paliaroutis, N. Sketopoulos, P. Tsoumanis, C. Sotiriou, N. Evmorfopoulos, and G. Stamoulis, "A Layout-Based Soft Error Rate Estimation and Mitigation in the Presence of Multiple Transient Faults in Combinational Logic," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*, 2020, pp. 231–236.

[17] F. A. Kuentzer and M. Krstic, "Soft Error Detection and Correction Architecture for Asynchronous Bundled Data Designs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–12, 2020.

[18] F. A. Kuentzer, M. Herrera, O. Schrape, P. A. Beerel, and M. Krstic, "Radiation Hardened Click Controllers for Soft Error Resilient Asynchronous Architectures," in *2020 26th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2020, pp. 78–85.

[19] F. A. Kuentzer and M. Krstic, "Assessing AFEDC Architecture's Robustness to Timing Faults," in *Proc. 33. GI/GMM/ITG-Workshop Testmethoden und Zuverlässigkeit von Schaltungen und Systemen (TuZ 2021)*, 2021, pp. 1–2.

[20] S. Beer, M. Cannizzaro, J. Cortadella, R. Ginosar, and L. L., "Metastability in Better-Than-Worst-Case Designs," in *International Symposium on Asynchronous Circuits and Systems*, 2014, pp. 101–102.

[21] M. Zamani and M. B. Tahoori, "A transient error tolerant self-timed asynchronous architecture," in *2010 15th IEEE European Test Symposium*, 2010, pp. 88–93.

[22] M. Anglada, R. Canal, J. L. Aragón, and A. González, "MASkIt: Soft error rate estimation for combinational circuits," in *2016 IEEE 34th International Conference on Computer Design (ICCD)*, 2016, pp. 614–621.

[23] A. Evans, M. Glorieux, D. Alexandrescu, C. B. Polo, and V. Ferlet-Cavrois, "Single event multiple transient (SEMT) measurements in 65 nm bulk technology," in *2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, 2016, pp. 1–6.

[24] M. Ebrahimi, H. Asadi, R. Bishnoi, and M. B. Tahoori, "Layout-Based Modeling and Mitigation of Multiple Event Transients," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 3, pp. 367–379, 2016.

[25] Y. Du, S. Chen, and J. Chen, "A Layout-Level Approach to Evaluate and Mitigate the Sensitive Areas of Multiple SETs in Combinational Circuits," *IEEE Transactions on Device and Materials Reliability*, vol. 14, no. 1, pp. 213–219, 2014.

[26] Y. Du and S. Chen, "A Novel Layout-Based Single Event Transient Injection Approach to Evaluate the Soft Error Rate of Large Combinational Circuits in Complimentary Metal-Oxide-Semiconductor Bulk Technology," *IEEE Transactions on Reliability*, vol. 65, no. 1, pp. 248–255, 2016.

[27] X. Cao, L. Xiao, J. Li, R. Zhang, S. Liu, and J. Wang, "A Layout-Based Soft Error Vulnerability Estimation Approach for Combinational Circuits Considering Single Event Multiple Transients (SEMTs)," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 6, pp. 1109–1122, 2019.

[28] M. R. Rohanipoor, B. Ghavami, and M. Raji, "Improving Combinational Circuit Reliability Against Multiple Event Transients via a Partition and Restructuring Approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 5, pp. 1059–1072, 2020.

[29] C. Georgakidis, C. Sotiriou, N. Sketopoulos, M. Krstic, O. Schrape, and A. Breitenreiter, "R-Abax: A Radiation Hardening Legalisation Algorithm Satisfying TMR Spacing Constraints," in *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2020, pp. 316–321.

[30] C. Georgakidis and C. Sotiriou, "Radiation Hardening Legalisation Satisfying TMR Spacing Constraints with Respect to HPWL," in *2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, 2020, pp. 1–6.

[31] C. Georgakidis, I. Lilitsis, G. Stanimeropoulos, and C. Sotiriou, "RADPlace: A Timing-aware RAdiation-Hardening Detailed Placement Scheme Satisfying TMR Spacing Constraints," in *2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, 2021, pp. 1–6.

[32] A. Peeters, F. te Beest, M. de Wit, and W. Mallon, "Click Elements: An Implementation Style for Data-Driven Compilation," in *International Symposium on Asynchronous Circuits and Systems*, 2010, pp. 3–14.

[33] P. A. Beerel, R. Ozdag, and M. Ferretti. Cambridge, UK: Cambridge University Press, 2010.