e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 2, April 2024

Impact Factor: 8.379

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

SEU Detection in FPGA Cram with Low Latency In-Memory ECC Checking

[1]Mr.T.Maheswaran, [2].Sathiyavani R, [3].Rubashri G, [4].Deepamalar J, [5].Sakthivel

[1]Assistant Professor, Dept. of ECE, MPNMJ Engineering College, Chennimalai, Erode, Tamilnadu, India

[2,.3,4,5] Final Year, Dept. of ECE, MPNMJ Engineering College, Chennimalai, Erode, Tamilnadu, India

ABSTRACT: In harsh environments such as space, radiation andcharged particles cause Single-Event Effects, faults occurring randomly on any electronic component. These must be mitigated to ensure device functionality. Modern mitigation methods, such as triple modular redundancy, are very effective against *Single- Event Transient s*(SETs),butincuraminimumof3×cost in area. *Single-Event Upsets* (SEUs) affect sequential elements and are regularly repaired using memory scrubbing. Scrubbing is a slow serial process, going through every memory word looking for errors to repair. It involves a non-negligible *Time To Detect* (TTD) before repair, during which other events can occur and compromise the system. *Field Programmable Gate Arrays* (FPGAs)rely heavily on sequential elements to store their config- uration; thus, FPGA's SEU detection time is critical to ensuring design integrity in harsh conditions. In this paper, we propose *In-Memory Error Code Correction Checking* (IMECCC),a method to replace memory scrubbing and improve FPGA configuration memory protection in high radiation environments. Our method allows asynchronous SEU detection, and replaces the scrubbing's variable time to detect with a fixed TTD. We show that IMECCC reduces FPGA's TTD by at least 116,000× on average, with an area increase of 1.56×, using a test architecture resembling a Xilinx Virtex 5 QV at a 60MHz scrubbing frequency.

KEYWORDS: Bit stream scrubbing, ECC, FPGA, MBU, MCU, partial reconfiguration, SEU, SET, TMR.

I. INTRODUCTION

FIELD-PROGRAMMABLE Gate Array's(FPGA) reconfigurability is an asset; it allows the modification of an implemented design, such as an upgrade or repair through routing and synthesis modification. This reconfigurable prop- erty is even more crucial in harsh environments like space, where radiation and ionizing particles increase potential for failure, and human maintenance is impossible or time- dependant. Indeed, some failures can be critical, causing damage to the hardware. used to restore the functionality of the device through recon- figuration; this is known as a fail soft capability. However, the *Configuration Random Access Memories* (CRAMs) that provide the fail soft capability are also vulnerable to events such as *Single-Event Upset* (SEU) and need to be protected and repaired to avoid unexpected FPGA behavior.

State-of-the-art *Radiation-Hardened*(Rad-Hard)parts, such as the Xilinx Virtex 5 QV [1] recently used in the NASA Per- severance Mars Rover, embed a combination of three methods to increase FPGA reliability. The first way to mitigate *Single- Event Effects* (SEEs) is physical hardening, such as replacing *Static Random Access Memory* (SRAM) with *Dual Inter- locked Storage Cell* (DICE-cell) and replacing conventional latches with *Radiation-Hardened-By-Design* (RHBD) ones [2], [3], [4]. The second method is physical redundancy, com- monly *Triple Modular Redundancy* (TMR), which triplicates the implemented design to decrease the likelihood of error via majority voting[5].Bitstream-scrubbing is the last of the mitigation methods, and entails serially traversing the CRAM words looking for SEU-induced errors to repair [6]. Scrubbing's serial nature induces a non-constant *Time To Detect* (TTD) because of the SEE's randomness during the serial verification process. It is possible for more than one SEU to occur before bitstream repair, creating a failure inspite of the redundancy and leading to erroneous behavior. Thus, reducing an FPGA's TTD after an SEU is critical.

In this paper, we introduce *In-Memory Error Correction Code Checking* (IMECCC), an interruption-based bitstream repair method that may be used in place of bitstream-scrubbing to reduce the FPGA TTD significantly. Our work presents a new FPGA architecture which asynchronously detects SEUs with a sensor network, and integrates *Error Correction Code* (ECC) checkers on every CRAM word. In the event of an SEU, a sensor triggers CRAM repair for its associated word. Compared to serial memory-scrubbing, IMECCC reduces the probability of an error being

compounded by a second SEU. Our work also shows a reworked configuration process that integrates ECC parity bits at the system level to avoid errors during the programming phase. Our method's approach has several benefits compared to commercial FPGAs. We demon-
strate a TTD reduction of at least 116,000× on average compared to a reference architecture resembling a Xilinx Virtex 5 QV at a cost of an area increase of 1.56× at the *Configurable Logic Block*(CLB)level compared to a similar XilinxVirtex5QV.

The rest of this paper is organized as follows: Section II provides technical background on SEE, mitigation methods, and FPGA customization; Section III details the IMECCC

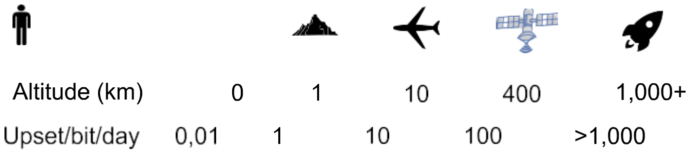


Fig.1.IllustrationofSEUperbitperdayatdifferentaltitudesfrom[7].design methodology, from the CRAM-cell design, to the self- protected extra hardware; Section IV develops the experimen- tal methodology and results; and Sections V and VI discuss the results and conclude.

II. RADIATION EFFECTS AND MITIGATION FOR FPGAS

This section briefly introduces the SEE s' impact on FPGA behavior and highlights the need for mitigation method imple- mentation. Then, short presentations of state-of-the-art meth- ods are provided including both soft and hard integration techniques. This section concludes with an introduction tostate-of-the-art Rad-Hard FPGA architecture.

A. Single-Events in FPGA

SEEs are radiation and ionizing particle-induced events, primarily present in harsh environments. Their main char- acteristics are: the *Linear Energy Transfer* (LET) as the amount of MeV-cm^2/mg transferred from the particle to the device, and the cross-section as the area of effect. SEEs have many sub-categories including SEUs and *Single-Event Transients* (SETs).Although their frequency can be estimated, as illustrated in Fig.1with the relation between altitude and SEU per bit per day[7],any given event's location and trigger time cannot be accurately predicted. An SEU may affect any sequential element in a design and flip the stored value. There are three types of SEU, as illustrated byFig.2:(1)*Single Bit Upset*(SBU)when only one memory bit is flipped,
(2)*Multiple Bit Upset*(MBU)when more than one memory bit is flipped in the same memory word [8], [9], and (3) *Multiple Cell Upset* (MCU) when more than one memory bit is flipped among two or more memory words [9]. SEUs are persistent errors and a considerable threat to FPGA design integrity; this is because CRAM cells are distributed throughout the FPGA area, and any alteration may lead to a design modification, such as a changed logic function or unexpected signal routing.SEUs will persist unless corrected; therefore, they require the implementation of a correcting system. SETs differ from SEUs in terms of error duration. SETs are temporary; the system restores itself to the original state when the event is passed. An SET duration mostly depends on the LET [10]. Moreover, SETs may affect any transist or in the design, generating analog variation leading to possible undetermined or unexpected logic states. SETs threaten FPGA functionality with effects such as signal alteration, unexpected set or reset, or early clock triggering.

B. State-of-the-Art FPGA Mitigation Methods

This section briefly introduces the state-of-the-art mitigation methods, starting with the hardware implementation in physical hardening. Then, the soft solutionsare presented in the TMR and bitstream scrubbing sub- sections.
Fig.2.IllustrationofSEU'simpact on two adjacent memory words where
(0) is the initial state,(1)shows an SBU,(2)shows a 2-bit MBU, and(3)shows a 3-bit MCU composed of a 2-bit MBU and an SBU.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

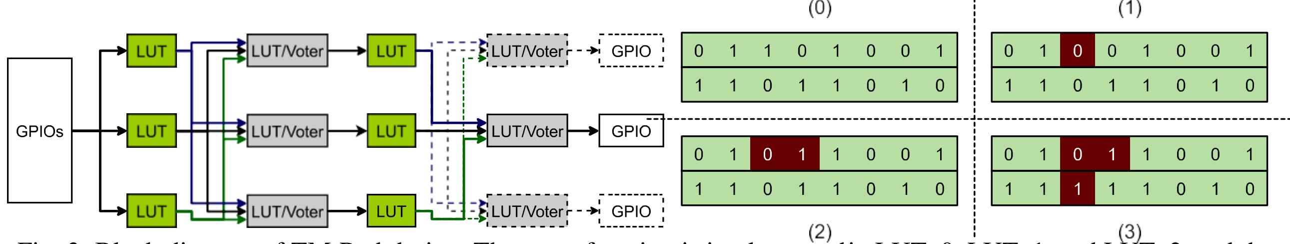


Fig. 3. Block diagram of TM Red design. The same function is implementedin LUT_0, LUT_1, and LUT_2, and the outputs are submitted to voters, transmitting the majority value to the remaining of the design.

1) Physical Hardening: The first level of physical hard- ening is technology selection. Previous work has shown that a fully depleted silicon on insulator *Static Random Access*

Memory (SRAM) is 6× more resistant to SEU than a bulk SRAM [11].The second level of physical hardening is RHBD.State-of-the-art Rad-Hard FPGAs implement RHBD latches
as CRAM [1]. RHBD latches can be either topologically different from standard latches or redesigned to break the cross-coupling between internal inverters [4]. SRAM, whichare also ubiquitous in modern FPGAs, can be hardened thesame way with DICE-cells [2]. RHBD techniques incur anarea overhead due to the larger size of RHBD cells comparedto non-hardened cells.

2) Triple Modular Redundancy: TMR is a very popularmitigation method that can be applied either at the system or design level. At the design level, TMR consists of designsegmentation and the triplication of those segments; the correct behavior for each output is determined by voters, as illustrated by Fig. 3. TMR protects the design from SETs and, in an FPGA, TMR delays the impact of SEUs on the CRAM [5]. Thus, TMR protects the design from CRAM bit-flip untilit is repaired or until another SEU happens on a redundant partof the design and makes it faulty. TMR's protection comes
with a high cost in FPGA utilization: 3× the original design, plus the voters. However, TMR does not have to be applied
to a complete design and could be partially applied to only themost critical parts to reduce the utilization overhead [12]. Other works replace TMR with sensor-based dual redundancy to reduce FPGA logic utilization [13].

3) Bitstream Scrubbing: Bitstream scrubbing is a repair method to recover the initial CRAM value after SEUs. Broadly, there are two scrubbing techniques in use [5]. The first and most simple is blind-scrubbing. A read-only memory or very radiation-resistant memory such as flash is embedded at the system level; the bitstream is continuously rewritten from the rad-hard memory. The second method is the most common and is called readback-scrubbing. It requires several modifications compared to blind-scrubbing: (1) ECC code integration to protect CRAM words against a defined number

of modifications,(2)ECC checker integration to interpret theECC status and trigger partial reconfiguration if needed, (3) address counters to load the ECC checker with differentCRAM words, and(4)logic around the ECC checker to triggerCRAM reconfiguration hardware. Read back-scrubbing goesthrough the bitstream, retrieves the data, checks the integritywith the ECC checker, and repairs the bitstream if necessary.This process takes time and implies a non-constant TTD, oftenreported as *Mean TTD* (MTTD). Some recent works replacethe counters with prioritization and specific algorithms or use high-speed ports to reduce the impact of SEUs [14], [15]. Furthermore, readback-scrubbing relies on the implementedECC. The Hamming code is mainly used and requires paritybit generation and integration, as illustrated in equation 1.The Hamming code is *Single Error Correction Double Error Detection*(SECDED)[16];itcanbedefeatedby3-bitormore errors, whose likeliness have recently been studied [17].

$$\#parity_bits = \log 2(\#data_bits) + 1 \quad (1)$$

4) Other Methods: Previous works have presented alterna- tives and improved solutions to readback scrubbing through detecting redundancies that trigger scrubbing on a definedportion of the bitstream or even decode the frame address [18], [19], [20]. As the bitstream scrubbing optimization introduced in the previous section, these solutions are architecture-,design-,and application-dependent. The application dependence refers to the probability for each CRAM bit to be involved and its bit flip detected. Therefore, the detection does not correlate directly with the time an event happens but with the time the affected bit would be used. Thus, such methods are not considered in this study.

C. State-of-the-Art FPGA Architecture

An FPGA is characterized as Rad-Hard when it includes some implementable or built-in mitigation methods, such as

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

those previously introduced, and passes irradiation tests, such as faster error recovery than particle rate or the Mil-PRF-38535 qualification. However, the architecture can vary, and in some cases, implementation details can present their own weaknesses. For example, the Xilinx Virtex 5 QV includes a SECDED ECC per configuration frame and RHBD latches as CRAM [21].A frame contains 1,312 bits divided into 41 mem-ory words. Such an organization minimizes the area overhead induced by the ECC bits but exposes the FPGA to potential simultaneous events. Therefore, physical hardening and risk assessment are required before utilizing such FPGAs in harsh environments. An alternative consists of integrating ECC bits in every CRAM word to reduce the ECC's probability ofbeing defeated. However, it implies a larger area-overhead than the frame solution. Both the frame and word solution are used as references in this work.

III. IMECCC FPGA ARCHITECTURE

State-of-the-art Rad-Hard FPGAs mainly rely on readback bitstream scrubbing to detect CRAM's SEU, which can take several microseconds. Such detection times are systems' lia- bility and limit the utilization of FPGAs for critical appli-cations. This work proposes a new FPGA architecture and

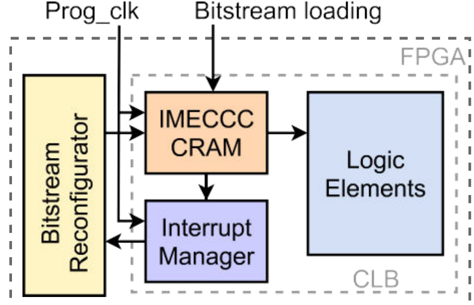


Fig. 4. IMECCC CLB block diagram illustrating the loading of CRAM words containing the ECC parity bits and the replacement of scrubbing with an interruption-based process.

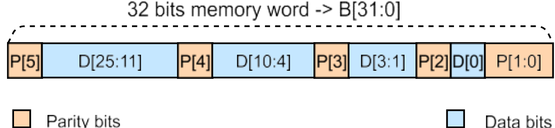


Fig. 5.Illustration of IMECCC CRAM word organizations with intertwined Hamming parity bits. A 32 bits CRAM word is composed of 26 bits of dataand 6 bits of parity.

design methodology to replace bitstream scrubbing with asyn-chronous SEU detection to reduce systems' liability. Our maincontributions are ECC and detection time improvements. Theyare obtained via ECC checkers integrated as SEU sensors with a specific floor plan which increases the ECC error coverage. This section describes the design methodology to integrate ourarchitecture on a fabric comparable to a Xilinx Virtex 5 QV, using the Google-Skywater 130nm *Process Design Kit* (PDK) as proof of concept. We first discuss the ECC implementation and its impact. Then, we present a memory design variation that eases the physical design and avoids any risk of unde- tected error. We then develop the use of these CRAM cells forin-memory ECC checking: how to floor plan these new CRAMs and use the ECC checking as a sensor to detect SEUs. Next we develop SEU sensing as a replacement for bitstream scrubbing offering more efficient protection. Finally we conclude this section with a self-protection analysis of the newly introduced hardware. A high-level introduction of the proposed work is illustrated by the block diagram in Fig. 4.

A. ECC Parity Generation and Coverage

The IMECCC architecture uses a *Single Error Correction Double Error Detection* (SEC-DED) Hamming code, also used in the reference FPGA. Previous work showed that SEUscould affect more than two bits[9],[17],and this concern is addressed in section III-C. In our architecture, we have
26 bits of configuration data (D)and 6 parity bits(P)in order to fit the 32 bit configuration word (B), as shown in Fig. 5 and according to equation 1. In our work, configuration data are protected when loaded to the FPGA and can be directly repaired if an event occurs. This CRAM organization differs from existing devices, which have 12 parity bits (P) for 1,280 bits of configuration data (D), corresponding to the frames introduced in Section II-C. The proposed design use the Hamming equations to protect every configuration data bit with at least two parity bits, as shown in

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

equations 2-6.

$$P[1] = \sum_{n=1}^{7} B[4\times n+2] \oplus B[4\times n+1] \quad (2)$$

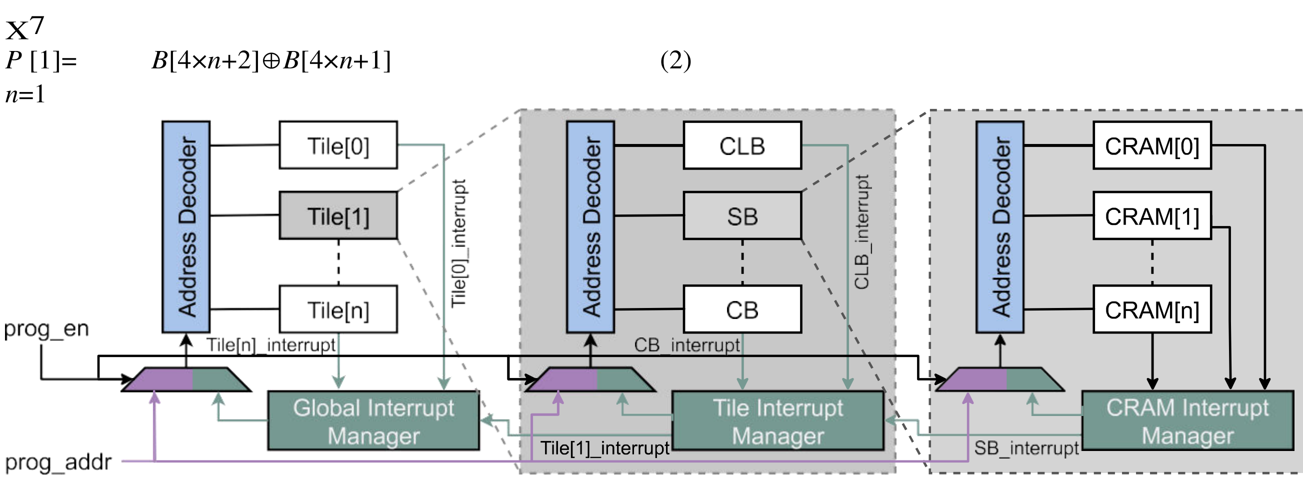


Fig.6. Illustrationofmemorydecoderscontrolledbysystem-levelsignalsprog_enandaddr_induringprogrammingphase(a)andmemorydecoderre-utilizationbytheinterruptmanagersinrepairphase(b).Both(a)and(b)aresequencedatprog_clkfrequency.

$P[2]=\oplus\{B[30:27],B[22:19],B[14:11],B[6:4]\}$ (3) $P[3]=\oplus\{B[30:23],B[14:8]\}$ (4)

$P[4]=\oplus B[30:16]$ (5)

$P[5]=\oplus B[30:0]$ (6)

B. Word-Based CRAM Access

SRAM-based FPGAs use latches as CRAM. Configuration can be loaded into the CRAM through many protocols, either serial or parallel. In this work, we use a word-accessible protocol that can reach every CRAM word from its unique address. This kind of BRAM-like organization is very useful for scrubbing and reconfiguration, easily reaching a memory word by its address and existing decoders. Indeed, as illus- tratedbyFig.6,the memory decoders are controlled at the system level during the programming phase and in by the interrupt managers described in Section III-D during the operating phase. More importantly, in the case of a SET happening on a clock signal, memories are not altered as long as the "write enable" signal is not set.

C. In-Memory ECC Checking

State-of-the-art Rad-Hard FPGAs rely on *Place and Route* (PnR) tools to automatically and efficiently distribute the CRAMs when designing the architecture. Thus, to minimize the risk for the Hamming code to be defeated, either the CRAM cells have to be as SEU resistant as possible to minimize the risk of ECC failure, or the integrated ECC has to be very resilient. Our work uses a specific floor plan illustrated in Fig. 7. This floor plan enables very fast SEU detection and reinforces ECC resilience by forcing MBU to beonly *Double Bit Error* (DBE).

1) SEU Detection: In commercial FPGAs, SEU detectionis mostly done by readback bitstream scrubbing. However, read-ing back the FPGA configuration looking for errors inevitably delays SEU detection. Indeed, it is very unlikely that SEUs will only happen at the next scrubbing addresses. State-of-the-art mitigation methods reduce this weakness with a com- bination of scrubbing and TMR, as mentioned in section II. Our work proposes an alternative to readback-scrubbing to combine with existing redundancy methods, such as TMR. We increase the quantity of ECC checkers to integrate one

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

TABLEI

SEU TYPE AND LOCALIZATION FROM ECC CHECKER VALUES

ECC[MSB], ECC[MSB-1:0]	Error Type
00	None
01	DBE
10	SBU on b[31] (P5)
11	SBU on b[ECC[MSB-1:0] - 1]

ECC checker per memory word. These ECC checkers are used as sensors to detect SEUs asynchronously. The ECC checkers are similar to those implemented in state-of-the-art FPGAs, following typical equations for the reference design and the proposed work. From the value of the ECC signal, both designs can determine the type of error and eventuallythe error location following the truth table illustrated in Table
I. One difference between the designs comes from equation 7used by our work to trigger the reconfiguration as explained in section III-D.

$$flag = |ECC \tag{7}$$

2) *ECC Error Coverage Improvement:* Improvement in the Hamming ECC error coverage is made possible by utilizing many ECC checkers as all or nothing sensors. Previous experimental work on MBU and MCU in BRAM showed that the majority of SEUs create SBU; two and three-bit errors are common enough to be considered, but alterations of four bits or more bits are very unlikely [9], [17]. A previous study showed SET duration and SEE cross-section increase as technology scales down, reaching upto1.56μm^2 at 100MeV- cm^2/mg LET for a 130nm technology node [10]. From these works and the dimensions of our CRAM, we defined an effective diameter of 6μm around a particle impactlocationtocoverLETupto1.8GeV-cm^2/mg and all potential trajectories. Then we created a memory word floor plan with a repeated pattern that ensures no more than two CRAM cells from the same word share any given 6μm space illustrated in Fig. 7. From this figure, we can see that ECC checkers are intertwined with CRAM cells. SEEs are represented with circles of diameter up to 6μm to illustrate SBU, MBU, and MCU. This figure also demonstrates that despite the Hamming code allowing only SEC-DED, our workreinforces the error coverage by preventing 3-bit MBUs.

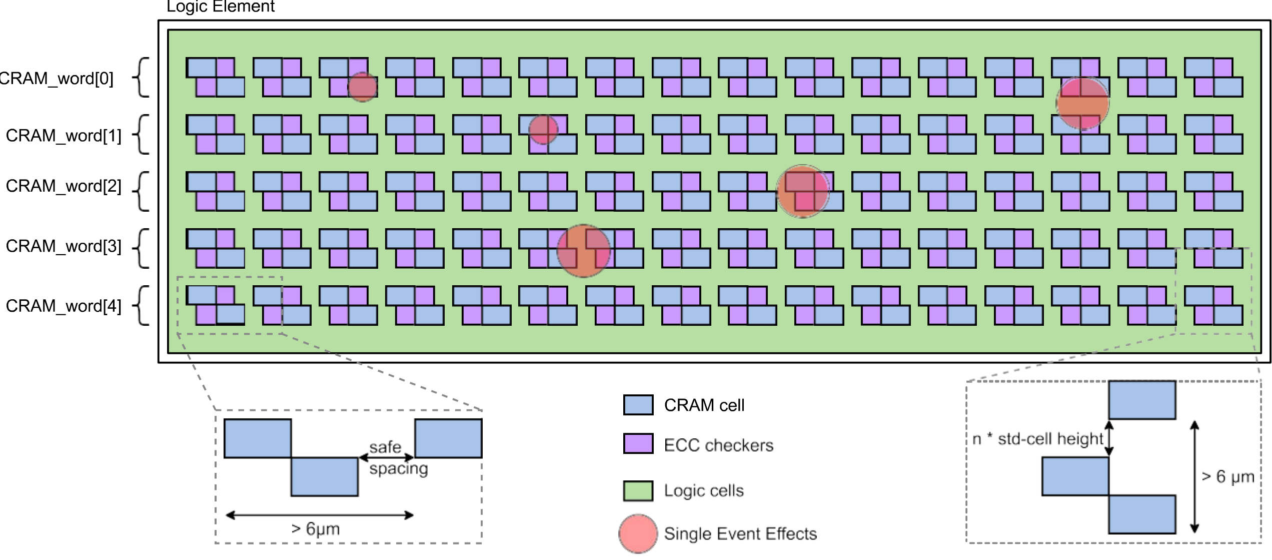


Fig.7. Representation of a Logic Element with In-Memory ECC Checking floor plan. It highlights the safes pacing of CRAM's cells and words with less than three CRAM cells in a 6μm diameter. The spacing is re-used for ECC checkers and logic integration. This figure also shows examples of SEE that can impact the CRAM.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

D. *Interruption-Based Reconfiguration*

Previous sections introduced the elements to enable the new interruption-based partial reconfiguration. This section presents the behavior of the interruption triggered by our solution. As this paper focuses on the detection time reduction through FPGA fabric customization, we intentionally avoid any reference to reconfiguration circuitry to preserve the FPGA designer repair time needs and customization possibilities, such as using existing self-repair methods or new ones in as many reconfigurable regions he wants, either internal or external to the FPGA. The combination of Hamming ECC code and IMECCC considerably reduces the risk of ECC failure. More importantly, IMECCC asynchronously detects SEUs in memory by creating a sensor for every memory word. These SEU sensors raise the "flag" signal from equation 7to trigger the interrupt manager in Fig. 4and 6(b). The interrupt manager contains a 3-bit counter for each CRAM word and prioritizes the memory words to repair and transfer to the bit stream reconfigurator. The counter's MSB is used as a time- out to prevent a destructive event or a Single Event Latch-up from freezing the entire system's reconfiguration capability. In the case of two or more SEUs happening in a very short period at different locations in the FPGA, the affected memory words are repaired one after the other. Interruption- based partial reconfiguration can repair the same memory word many times consecutively in the case of many SEUs affecting this same memory word, increasing the reliability of the proposed work over the current state-of-the- art. Interruption-based and scrubbing-based partial reconfiguration share the same reconfiguration process as illustrated by the flow chart in Fig. 8. This figure highlights in yellow the limitation of Hamming-based reconfiguration and the cost of double-error detection, which requires system intervention to repair the affected memory word.

E. *Self-Protected Hardware*

Although IMECCC provides protection against SEEs, itdoes so by introducing new hardware, which itself may be

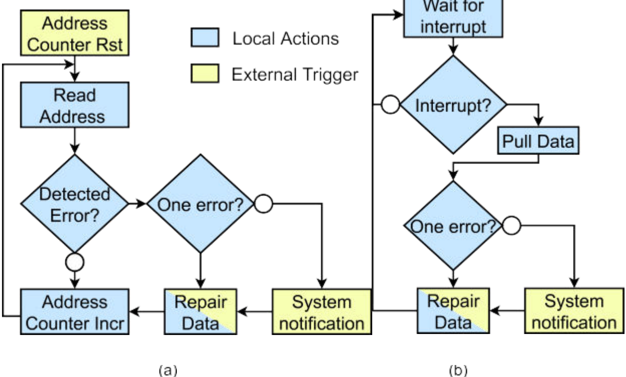


Fig.8.Flow- chart presenting the reconfiguration difference between (a) read- -back scrubbing and (b) the proposed work, as well as the internal or external location of the involved elements.

subject to SEEs. In this section, we explain how the sensors and interrupt manager are self-protected from SEUs and SETs.

1) *Sensors:* The sensors are composed of combinational gates; thus, they have total immunity to SEUs. Although ECC checkers are not TMRed, SETs do not threaten the system's behavior. Indeed, a SET happening on the sensor's logic will trigger an interruption for the event duration at most. In the case of a long duration SET, two cases are possible: (1) the SET affects ECC signals involved in ECC [4:0], creating a false DBE detection and reprogramming this memory word byloading the correct value from the system-level until the SET ends, as shown in Fig.9.Or,(2)the SET also affects the fifth bit of the ECC signal, and falsely targets a bit of the memory word for reprogramming until the SET ends. Then the last reconfiguration ensures the correct value is restored. SETs canalso affect the logic gates generating the "flag" signal. Then, two behaviors are possible:(1) a gate computing

2) *Interrupt Managers:* The interrupt managers are not immune to SEU because of the embedded counters for timeout

International Journal of Innovative Research in Computer and Communication Engineering

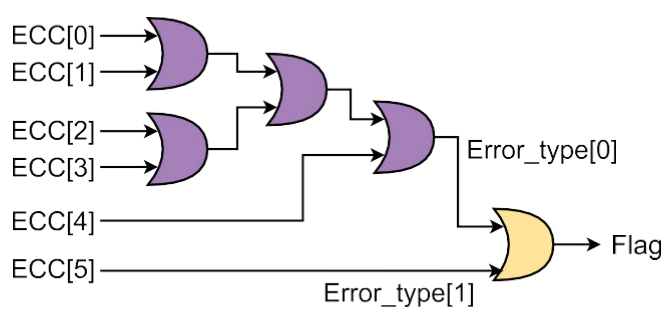
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India



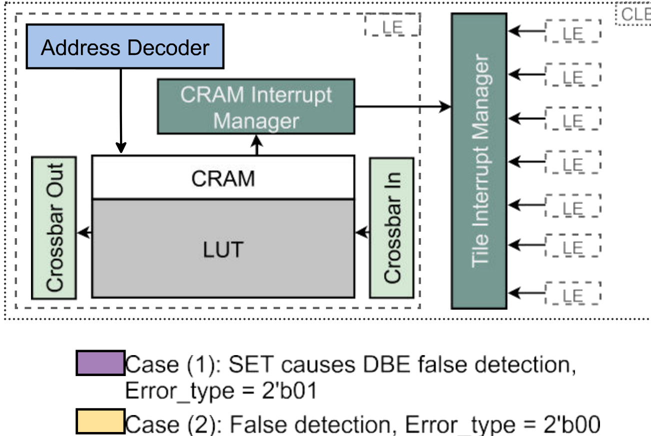


Fig.11. Simplified CLB schematic view showing the inter connectionbetween all the presented modules with a zoom inside an LE.

Fig.9.Schematic of implemented logic for the flag and error_ type signals. Purple gates are potential causes of DBE detection caused by SET, as definedin case (1). SET on the yellow gate causes false detection without reconfiguration, as in case (2).

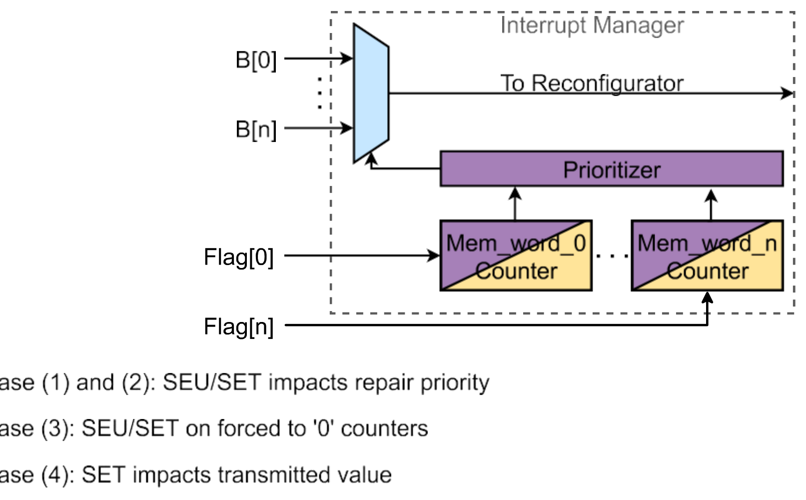


Fig.10.Interrupt manager block diagram to identify SEUs and SETs target. Purple blocks are impacting repair priorities because of SEU and SET, as defined in cases (1) and (2). SEU/SET on the yellow elements has no impact on forced to '0' counters, as in case (3). The blue multiplexer named case (4) shows how memory bits can be incorrectly transmitted.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

mentioned in section III-D. SEUs can occur on the interrupt managers in three different ways, as depictedinFig.10: (1) It increases the counter value and indirectly the associated element priority. Eventually, it triggers a system notification of hardware potentially altered; (2) It decreases the counter value and the associated element priority. System notification is also delayed; Or (3) the SEU happens on an element's counter which is not already in error. Counter's value is forced to '0,' and the event acts as a SET. SETs can happen on the multiplexers carrying the signal from a memory word to the reconfigurator. As for the sensor error, it leads to reconfiguration during the event and a correction once the event is over. However, this case only happens if a SET hits

LEs (k6N8 architecture) and are designed in a hierarchical flow floor planed as illustrated in Fig. 11.

A. Methodology

We used Open FPGA to generate the reference designs and our work on IMECCC [22]. Open FPGA is an open-source FPGA prototyping tool that takes an XML description of the architecture to implement and the associated cells from a PDKas input to generate the equivalent net lists ready for physical design in a semi-custom flow. In this work, it was used with an architecture resembling a Xilinx Virtex 5 QV adapted to the Google-Sky water 130nm PDK to generate the reference designs (*Refframe* and *Refword*). The CRAM cells are imple- mented as non-hardened latches. The configuration CRAM organizations are respectively frame-based for *Refframe* and word-based for *Refword*. Our work (*IMECCCCLB*) is a mod- ified version of *Refword*. It includes net list modification and specific floor planning of the IMECCC CRAM, as described in section III. Every module added to the net lists was synthe- sized, and technology mapped using Synopsys Design Com- piler. Functional verification for each module was done with the Mentor Graphics Questa sim simulation tool and manual test benches. The physical design was realized using SynopsysICC2. Experimental results provided in the next sections focus on

the CLB area and TTD comparison. The area comparison was performed after physical design .*Refframe* TTD (*MTTDframe*) is determined as half the readback CRC scan time to emulate a gaussian distribution of SEUs[23]. *Refword*

TTD (*MTTDword*) is calculated as an average based on the Xilinx Virtex 5 QV number of configuration frames —3,762 to63,024—at20%,40%,60%,and80%utilizationwith a scrubbing frequency of 60MHz, the Xilinx Virtex 5 QV recommended scrubbing frequency [23]. *MTTDword* follows equation8. *IMECCCCLB* TTD(*TTDIMECCC*) is extracted post- layout using Synopsys Prime Time as the longest time for an IMECCC CRAM word to raise its error flag.

an interrupt manager while a covered memory word is in thequeue for reconfiguration.

$$MTTD_{word} = \frac{\#word(frame)\times utilization(design)}{frequency\ (scrub_\ clock)\times 2} \quad (8)$$

V. EXPERIMENTALRESULTS

This section focuses on comparing CLBs' and *Logic Ele- ments'*(LEs) areas as well as SEU detection time between the proposed IMECCC architecture and reference designs with CRAM accessible by frame or word. All CLBs contain eight

B. Area Comparison

The physical design experiments showed areas of 84,441μm^2, 85,201μm^2, and 127,517μm^{2}at CLB level for *Refframe* *Refword* and *IMECCCCLB* respectively. Suchresultscauseanareaoverheadfromourworkof51%and

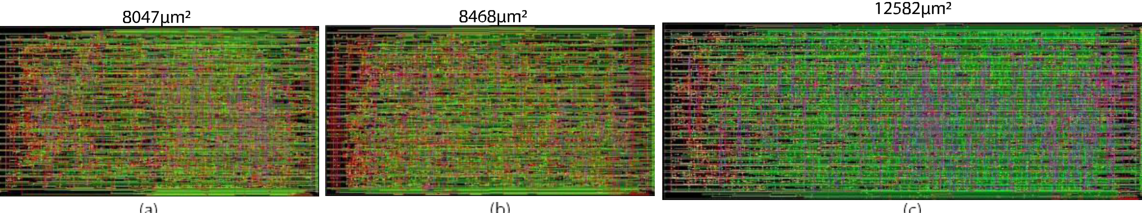


Fig. 12.LE's post-PnR views of: (a) *Refframe*, (b) *Refword*, and (c) *IMECCCCLB*. These views highlight the area overhead between the design caused by the CRAM organization. *Refword*is 5% larger than *Ref frame*, while *IMECCCCLB*'s LE is56% and 49% bigger than *Refframe* and *Refword*respectively.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

50% compared to *Refframe* and *Refword*. At LE level, block areas are 8,047μm^2, 8,468μm^2, and 12,582μm^2 for *Refframe Refword* and our work respectively, as illustrated in Fig.12. At this level, the area overheads reach 56% compared to *Refframe* and 49% compared to*Refword*. Such overhead is explained by the extra hardware from the locally integrated ECC checkers and interrupt-managers. However, this CLB to CLB does not consider the required FPGA hardware utilization to implement readback-scrubbing for the reference designs. The scrubbing- related utilization depends on the implemented benchmark, the scrubbing frequency, and the algorithm used. Such considerations are discussed in section V.

C. TTD Improvement

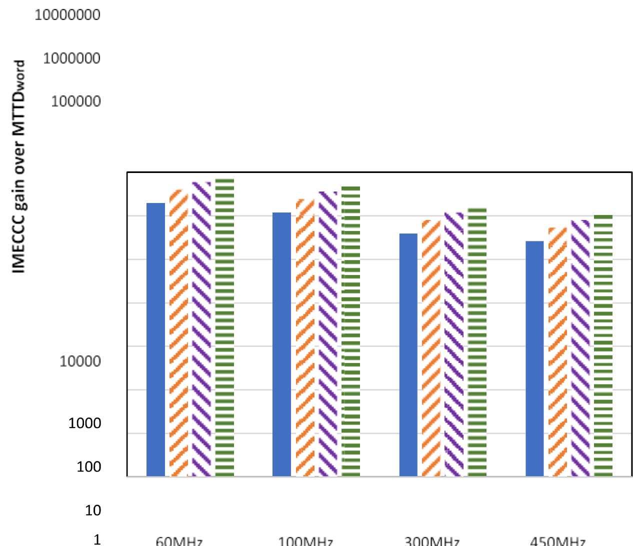
State-of-the-art SEU mitigation methods for FPGAs have shown a limitation in detection time. Therefore, the IMECCC technique has been designed to detect SEUs faster than readback-scrubbing to minimize this weakness. Experiments demonstrated an IMECCC sensor's TTD up to 2.16ns. In the meantime, calculations show that *MTTDframe* is between 1.27ms and 18.4ms depending on FPGA model[23].It impliesa detection time reduced by a factor of 587,000× to
8,518,000× for FPGA sizes similar to Xilinx Virtex 5 QVLX20T and LX330T respectively. The same FPGA sizes are used in the experiments comparing *MTTDword* with *TTDIMECCC*. As illustrated in TableII, IMECCC reduces SEU detection by 116,000× on average at the size of the LX20T and 20% utilization. Comparison with a *Refword* FPGA sized as the LX330T showed even larger improvement.

As illustrated in Fig.13,IMECCC improvement exceeds

TABLEII COMPARISIONBETWEEN*IMECCCCLB*'STTDAND CALCULATED*MTTDword*FORANFPGASIZED SIMILARLYTOXILINXVIRTEX5QVLX20T

Util	$MTTD_{word}$ (ns)	TTD_{IMECCC}	MTTD reduction
20%		2.16	
40%		2.16	
60%		2.16	
80%		2.16	

250,800	116,111×
501,600	232,222×
752,400	348,333×
1,003,200	464,444×



20% utilization 40% utilization 60% utilization 80% utilization

Fig. 13. Logarithmic bar graph showing the *IMECCCTTD* gain over *MTTDword* for scrubbing frequency between 60MHz and 450MHz and logic utilization varying between 20% and 80% at Xilinx Virtex 5 QV LX330T size.

56% increase in CLB's area for a k6N8 architecture and32-bit CRAM words. Overall, the IMECCC SEU detection method provides an *Area-Detection Product Improvement* (ADPI),
detailed in equation 9, between 78,146× and 3,022,381×.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

area (RefCLB)×MTTDRef (Util.,Freq)

1,000,000× for utilization over 20% at 60MHz scrubbing. Thus, we investigated how the performances would evolve,*ADPI* =

(IMECCC

CLB)×TTD

IMECCC (9)

we could scrub at frequencies up to 450MHz, which is XilinxVirtex5QVmaximumoperatingfrequency[1].IMECCC remains 250,000× faster than scrubbing for an FPGA with 20% utilization scrubbed at 450MHZ. However, the MTTD does not represent the worst-case scenario for readback scrub- bing. Indeed, in the scrubbing worst-case, an SEU happens on the very last CRAM word to check. In such conditions,

scrubbing has a TTD upto 9,700,000×slower than IMECCC. Furthermore, scrubbing's SEU detection best-case, occurring

on the next CRAM word to verify, remains 1.03×at 450MHz and up to 7.7× at 60MHz slower than our solution. The improvement from the IMECCC comes from an entire system

modification compared to state-of-the-art FPGAs. Indeed, the sensor network asynchronously detects SEUs, while scrubbingis a clocked step-by-step "random" search.

The IMECCC solution comes with a 2.16ns fixed TTD,

significantly reducing the detection time by at least 116,000× compared to scrubbing's MTTD. It also comes with an upto

V. DISCUSSION

The IMECCC solution presented in this paper is an uni- versal built-in detection technique, whose characteristics are summarized in Table III, that fundamentally differs from the implementable state-of-the-art solutions. Therefore, this solution comes with extra hardware to implement the sensors and manage interruptions and reprogramming compared to state-of-the-art FPGAs. For example, the interrupt-manager would use 150 LUT and 96 FF if implemented in reconfig- urable logic, which is comparable to the scrubbing methods mentioned in previous work [14]. This extra hardware, as well as the new approach, allows a significant gain in design implementation time. Indeed, IMECCC natively provides CRAM supervision and does not require modifications to implement the scrubbing algorithm.

Nonetheless, the IMECCC area overhead does not increase the probability of error. Eventhough a larger area increases

TABLEIII

SUMMARY OF THE PARAMETERS OF INTEREST COMPARISON BETWEEN STATE-OF-THE-ART FPGA AND THE IMECCC PROPOSED SOLUTION

Parameter	State-of-the-Art	IMECCC
Relative CLB area	1	1.52
RHBD	Founder Choice	User Choice
Redundancy	User Implementation	User Implementation
Detection Time	Average time in µs	Fixed time in ns
Detection Method	Custom Implementation	Built-in
Repair Method	Founder IP	Universal
Self-Protected	No	Yes

the likelihood of an SEE happening in a module, simulations have shown the hardware's resilience to both SEUs and

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

SETs.Indeed, most of the side effects of events on the new elements lead to reloading a correct bitstream in to a non-altered CRAM; otherwise, the side effects are neutralized in the absence of SEU affecting the configuration memory.
Moreover, this work considers only the area increase in the CLBs; the overhead for the routing modules should be lower since their areas are less CRAM-dependent. Furthermore, it should be possible to improve utilization performance by exploring alternatives to TMR that interruption-based partial reconfiguration may enable. For example, recent work has demonstrated that full TMR may require too many resources for reliability improvement compared to *Partial TMR* (PTMR)[24]. Some other recent works, such as [13]and[25],have also shown that dual redundancy can save up to 33% utilization compared to TMR and PTMR. Such ameliorations would directly benefit the ADPI and make the IMECCC more attractive.
The improvement from the IMECCC architecture comes with a cost in area at the CLB level when compared to the reference design. Such overhead comes from the staggered row floor plan and the extra logic. We estimated the proportional area overhead to remain approximately constant for technology nodes down to 12nm, and more advanced nodes may require more customization. However, FPGA area performance is not limited to element size, and the utilization ratio plays a significant role. Readback-scrubbing can be performed in many ways, following different algorithms and priorities depending on the environment and the implemented design, impacting FPGA utilization. The 52% increase in element'sareaisestimatedusing32-bitCRAMwordsandcan bereduced by implementing wider CRAMs. For example, a 64- bit CRAM word contains six parity bits, while a 32-bit CRAMword needs five of them. Modifying the CRAM word width does not only affect the number of parity bits; it also affects the number of CRAM words, the required logic for the ECC checkers, and the FPGA pin out. Hence, this sizing has to be considered and efficiently handled. Future work may investigate how to serially load FPGA configuration in a self- protected way in order to maximize the number of I/Os.
Another topic of interest is power consumption. Readback- scrubbing consumes significant quantities of dynamic powerby constantly loading the ECC checker, and this effect is evenmore pronounced when the FPGA implements high- speedscrubbing. IMECCC principally consumes static power

because the reconfiguration is triggered only by effective SEU/SET. It may be interesting, then, to investigate if IMECCC's benefits extend to power consumption.
Finally, the IMECCC solution provides many promising characteristics, which can be furtherly assessed in future works, on top of its very short TTD. Such improvement increases system reliability for existing applications in exoat-mospheric and nuclear environments. Thus, it is likely that the IMECCC architecture may allow new applications that current state-of-the-art components are unsuitable for.

VI. CONCLUSION

This paper presents a built-in solution to detect SEUs inFPGAs' configuration memory asynchronously called IMECCC. It is a new FPGA architecture using SEU sensors built from ECC checkers as an alternative to bitstream scrub-bing to quickly and asynchronously detect bit-flip in FPGACRAM. It extends sensor utilization in a sensor network,covering every CRAM word and enabling interruption-basedconfiguration repair to be integrated. Such a fabric-based approach has the advantage of allowing existing soft-IP to be implemented without spending an undertaking costly andtime-consuming development of a dedicated CRAM veri- fication algorithm. This detection method demonstrates an improvement between 116,000× and more than 9,700,000× TTD reduction compared to the MTTD for read-back scrub- bing. Such significant results are made possible by replacing
the serial process from the state-of-the-art scrubbing method with the IMECCC sensors, fixing the TTD to 2.16ns. Further-more, our solution uses a specific floor plan pattern, which,
along with the CLB design, shows the capability to reinforce integratedECCcodes.Inthiswork,thethreatofa3-bit MBU being undetected by the Hamming code is removed by integrating sensors inside the memory words, increasing the spacing of the CRAM cells enough to make a 3-bit MBU extremelyunlikely. These improvements are made at the cost of a
1.56× increase in area. Despite the cost in area induced by thesensor integration, the IMECCC solution provides an ADPI
between 78,146× and 3,022,381× compared to state- of-the- art SEU mitigation methods for FPGAs.

ACKNOWLEDGMENT

This paper was produced by the Laboratory for Nano Inte- grated Systems (LNIS),The University of Utah, USA. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not with standing any

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

copyright notation thereon. The views and conclusions contained here in are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) or the U.S. Government.

REFERENCES

1. *Radiation-Hardened, Space-Grade Virtex-5QV Family Data Sheet: Overview*, DS192 (V1.6). Xilinx, San Jose, CA, USA, 2018.
2. S. Whitaker, J. Canaris, and K. Liu, "SEU hardened memory cells for a CCSDS Reed– Solomon encoder, "*IEE Trans. Nucl. Sci.*, vol. 38, no. 6, pp. 1471–1477, Dec. 1991. [Online]. Available: https://ieeexplore.ieee.org/document/124134/
3. J. Kim and P. Mazumder, "A robust 12T SRAM cell with improved write margin for ultra-low power applications in 40 nm CMOS,"*Integr., VLSI J.*, vol. 57, pp. 1–10, Mar. 2017. [Online]. Available:https://linkinghub.elsevier.com/retrieve/pii/S016792601630070 0
4. B. Li and D. C. Brinkley, "Single event upset (SEU) hardened latch circuit," U.S. Patent 6327176, Apr. 2001.
5. M.Wirthlin,"High-reliabilityFPGA-basedsystems:Space,high- energyphysics, and beyond," *Proc. IEEE*, vol. 103, no. 3, pp. 379–389, 2015.
6. H. Carmichael and P. E. Brinkley, "Techniques for mitigating, detecting, and correcting single event upset effects in systems using SRAM-based field programmable gate arrays," U.S. Patent 7512871,Mar. 2006.
7. Petersen, *Single Event Effects in Aerospace*. Hoboken, NJ, USA: Wiley, 2011, ch. 13.
8. M. Wirthlin, D.Lee, G. Swift, and H. Quinn,"Multi-cell upsets within the Xilinx series-7FPGAs,"BrighamYoungUniv., Provo, UT, USA, Tech. Rep., 2014. [Online]. Available: https://www.osti.gov/servlets/purl/1140717
9. M. Ebrahimi, P.M.B. Rao, R. Seyyedi,and M. B.Tahoori,"Low-cost multiple bit upset correction in SRAM-based FPGA configurationframes,"*IEEETrans.VeryLargeScaleIntegr.(VLSI)Syst.*,vol.24,no. 3, pp. 932–943, Mar. 2016.
10. B. Narasimham et al., "Characterization of digital single event transient pulse-widths in 130-nm and 90-nm CMOS technologies," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 6, pp. 2506–2511, Dec. 2007.
11. W. Calienes, R. Reis, C.Anghel, and A.Vladimirescu, "Bulk and FDSOISRAM resiliency to radiation effects," in *Proc. IEEE 57th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2014, pp. 655–658.
12. B. Pratt,M. Caffrey, J.F. Carroll, P. Graham, K. Morgan, and
13. M. Wirthlin,"Fine- grain SEU mitigation for FPGAs using partial TMR,"*IEEE Trans. Nucl. Sci.*, vol. 55, no. 4, pp. 2274–2280, Aug. 2008.[Online]. Available: http://ieeexplore.ieee.org/document/4636895/
14. A. Alacchi, E. Giacomin, X. Tang, and P.-E. Gaillardon, "Smart- redundancy: An alternative SEU/SET mitigation method for FPGAs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.
15. He, S. Zheng, and N. Jing, "A hierarchical scrubbing technique forSEUmitigationonSRAM-basedFPGAs,"*IEEETrans.VeryLargeScaleIntegr.(VLSI)Syst.*,vol.28,no.1 0,pp. 2134–2145,Oct.2020.[Online].
16. Available:https://ieeexplore.ieee.org/document/9162497/
17. Stoddard, A. Gruwell, P. Zabriskie, and M. Wirthlin, "High-speed PCAP configuration scrubbing on Zynq-7000 all programmable SoCs,"in *Proc. 26th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2016.
18. *Single Error Correction and Double Error Detection*, XAPP645 (v2.2),Xilinx, San Jose, CA, USA, 2006.
19. J. Furuta, K. Kobayashi, and H. Onodera, "Impact of cell distance and well-contact density on neutron-induced multiple cell upsets," in *Proc.IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2013, pp. 298–303.
20. C.Bolchini, A.Miele, and C.Sandionigi, "A novel design methodology for implementing reliability-aware systems on SRAM-based FPGAs,"*IEEE Trans. Comput.*, vol. 60, no. 12, pp. 1744–1758, Dec. 2011.
21. L.Pereira-Santos,G.L.Nazar,andL.Carro,"Exploringredun- dancygranularitiestorepairreal-timeFPGA-basedsystems,"*Micropro- cess. Microsyst.*, vol. 51, pp. 264–274, Jun. 2017. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S0141933117 302272
22. M. Mousavi, H. R. Pourshaghaghi, H. Corporaal, and A. Kumar,"Scatterscrubbing: A method to reduce SEU repair time in FPGAconfigurationmemory,"in*Proc.IEEEInt.Symp.DefectFaultToleranceVLSI Nanotechnol. Syst.*, Oct. 2019, pp. 1–6.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details