DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING KANCHEEPURAM
CHENNAI - 600127

Synopsis Of

Fault Tolerant Control Path Hardware Design for Space Applications

A Thesis

To be submitted by

DEEPANJALI.S

For the award of the degree

Of

DOCTOR OF PHILOSOPHY

1 Abstract

The operation of digital systems in Lower Earth Orbit (LEO) faces increased fault risks from ionizing particles, impacting memory components, and causing bit flips known as Single Event Upsets (SEUs). Mitigating SEUs in control path hardware is critical, as a bit-flip can lead to erroneous control signals and operational failure. The control path hardware contains two design approaches, a HardWired Control circuit (HWC) and a Micro-Programmed Control Unit (MPCU), differing in how control signals are generated. Mitigation approaches for both implementations are necessary to address SEU and Multiple-Bit Upset (MBU). To meet the specific requirements of the control circuit design approaches, a distinctive fault mitigation approach is used for MPCU and HWC. For MPCU, the proposed information redundancy-based Error Correction Code (ECC) approach addresses challenges like inconsistent code rates and soft decoding logic that are highlighted in related works. In HWC, fault recovery is exponential, requiring a bio-inspired Evolutionary Algorithm (EA) approach for effective transition from faulty to non-faulty circuits. Virtual Reconfigurable Circuit (VRC) technology is accompanied by Evolvable Hardware (EHW) for FPGA-generic reconfiguration and mitigation of both SEU and MBU in the control circuit. Extensive fault simulations reveal potential requirements for fault mitigation within the EA module, leading to a proposal for a three-layer approach: *VRC implementation of control circuit*, faults corrected with *EA modules*, and EA module faults mitigated by an additional layer inspired by *Immunotronics*, ensuring comprehensive fault mitigation in the control circuit.

2 Objectives

- Analyze the control circuit's operational characteristics with applications related to space-based design, emphasizing its interaction with other system components to determine its mission criticality.
- Perform an extensive simulation model for SEU and MBU on the control circuit to characterize faults and evaluate their impact on the digital system.
- Investigate traditional fault mitigation approaches applied to the control circuit, delving into the challenges they encounter.
- Devise a unique fault mitigation strategy targeting specific variations observed in the design approaches of the control circuit. Evaluate this approach extensively using standard metrics like resource utilization, fault recovery time, and fault mitigation efficiency.

3 Existing Gaps Which Were Bridged

The gaps existing in the redundancy techniques and evolvable hardware techniques are as follows:

- **Inconsistent Code Rate in ECC Codes:** The widely employed information redundancy for SEU mitigation is the Hamming code. However, a primary challenge associated with this method is the code rate. The redundancy rate, determined by adding parity bits, escalates with increasing data length. For instance, the redundancy rate for Hamming(4,7) is 0.571, and it rises with the augmentation of data bits, reaching 0.969 for Hamming(255, 247) as it is demonstrated by Kim *et al.* (2007). While the Hamming code effectively addresses SEU, our research extends to devising an ECC for MBU.

- **Complex ECC-based Decoding Logic for MBU:** A prevalent challenge identified across various MBU codes is incorporating *soft decoding* logic. This entails employing probability or likelihood measures for classifying each bit as faulty or non-faulty. However, this approach results in heightened hardware complexity during the implementation of decoding logic, as stated by Hailes *et al.* (2015). Also, SEUs cause sudden, sporadic errors, challenging soft decision codes due to their reliance on continuous values and probabilistic models, making it difficult for these codes to handle the unpredictable, discrete errors of SEUs without specific patterns; thus, in SEU-prone settings, alternative error correction methods like hard decision codes or specialized mechanisms are recommended over soft decision codes.

- **Elementary Circuit Consideration in EHW:** The related work clearly illustrates that the circuits evolved and mitigated from faults possess a lower complexity. Complex circuits are not considered because of higher evolution time when bitstream evolution is considered. A smaller circuit, like a 3-bit multiplier, constitutes the lengthier chromosome of 431 bits as demonstrated by Wang and Liu (2017). Although circuits are less complex, the description regarding the number of configuration bits is high. Hence, smaller circuits were only considered.

- **Non-Generic Reconfiguration Technique:** The evolution of configuration bit is not possible in non-commercial FPGAs specifically designed for military and security-related applications. The Dynamic Partial Reconfiguration (DPR) tools and ports for accessing the configuration bit are unavailable in former FPGAs. Also, in non-commercial FPGA, bitstream access is prohibited by encrypting the configuration bit. The tools available to access the bitstream in commercial FPGA are FPGA-vendor specific, as demonstrated in the works of Trimberger *et al.* (2011).

- **Expectancy of SoC:** Most of the works such as Lohn *et al.* (2003), Wang and Liu (2017), Zhu *et al.* (2017), Silva and Duarte (2018) rely on the softcore processor for conducting the evolution process. The advantage of this methodology is that it allows the user to encode genetic algorithms in a high-level language like C, contrary to Hardware Description Language (HDL). However, communication of phenotype and genotype can decrease the fault mitigation speed.

- **Extensive Architecture for VRC Implementation:** The two-dimensional programming elements (PE) architecture in VRC is fundamental for emulating circuit functionality with multiplexers. However, its size is comparatively more significant than the standard implementation. As the architecture serves as the phenotype of EHW, increasing the number of PEs can enhance the genotype and augment the search space. For instance, even for a less complex circuit like a 3-bit multiplier implemented by Sekanina and Friedl (2004), the VRC produces eighty PE owing to 880 bits of configuration.

- **Non Partial Genotype Representation** In the DPR methodology for FPGA design, faults occurring in the configuration memory can be effectively isolated by breaking them down into frames. A portion of the bitstream can be reconfigured for the circuit to regain functionality. However, regarding reconfiguration based on VRC, the configuration bit stored in the configuration register mimics the configuration memory bitstream. Despite the usefulness of this approach, partial configuration or fault localization still needs to be explored. Furthermore, in the event of a single event upset, all the configuration bits are evolved, resulting in an insignificant evolution as performed by Zhu *et al.* (2017).

- **Unaccelerated Convergence** Convergence or fault recovery time is a critical metric for assessing the effectiveness of fault mitigation in terms of generation. Its value can be affected by various factors, including the method of phenotype representation, the efficacy of selected genetic operators, and the search space size. In mission-critical applications, convergence is a crucial parameter. Most of the circuit's convergence speed is low since the above two challenges have yet to be confronted.

4 Most Important Contributions

The main contribution of our research is:

- **Contribution 1:** A lightweight encoding and decoding logic for fault mitigation approach in MPCU architecture:
  - The primary objective of our proposed system is to design a hard decision decoder for mitigating both SEU and MBU. Additionally, we address the challenge of inconsistent code rates, focusing on achieving a consistent code rate of length $\frac{n}{2} + 2$ using the self-inverting property of XOR Gates.

- **Contribution 2:** Fault Mitigation approach for HWC approach using evolvable hardware:
  - A self-healing-based approach is utilized for fault mitigation in the HWC-based design of the control circuit. The main contribution to this objective is achieving a compact virtual reconfiguration and constrained evolution to accelerate the fault mitigation process through the intrinsic implementation of an EA.

- **Contribution 3:** A comprehensive fault mitigation approach to safeguard evolvable hardware from faults:
  - The solution mentioned above integrates genetic algorithm modules intrinsically onto the same reconfigurable fabric used for HWC-based control circuit design, aiming to minimize communication delays. However, this integration has compromised the EA's reliability, leading to potential faults within the genetic algorithm modules. Therefore, an additional protective layer is introduced aiming at fault mitigation for the genetic algorithm modules, operating independently from the control circuit's functionality.

4.1 Fault Mitigation Approach in MPCU architecture:

Our proposed ECC scheme adopts the self-inverting property of the XOR gate, demonstrated through equations (1), (2), and (3). These equations depict the XOR operations between bit-strings, showcasing how deviations in bit-strings (D) of length n can recover the original bit-strings ($B1$ and $B2$) through reversible XOR operations.

$$B1(n) \oplus B2(n) = D(n) \tag{1}$$

$$D(n) \oplus B2(n) = B1(n) \tag{2}$$

$$D(n) \oplus B1(n) = B2(n) \tag{3}$$

4.1.1 Proposed SYMmetric SEGmented (SYM SEG) codes for Fault Mitigation in MPCU

The self-inverting properties of the XOR gate prompt its adoption in our fault mitigation design for the MPCU control word encoder and decoder hardware logic, nevertheless, with adjustments for the control word's organizational structure. While normally requiring a pair of bit-strings, the control word provides only one at a time, leading to a solution involving symmetric segmentation to utilize this property. This segmentation retains the property's integrity as the segments function as bit-strings, enabling their application for fault mitigation. The proposed architecture involves encoding by segmenting control words, computing check bits, and adding parity bits for fault detection. Decoding relies on observed parity changes, using XOR operations to calculate expected control word segments based on complementary segments and check bits. The control store is depicted in Figure. 1 illustrates the MPCU architecture's control memory in its stand-alone form, lacking check bits. This store comprises a matrix with m rows, each containing control words of length n. After an encoding process, this store is divided symmetrically at each row into two segments, $CW[0 \dots n/2-1]$ and $CW[n/2 \dots n-1]$. Subsequently, parity bits ($P1$ and $P2$) are appended, followed by check bits (Cb) added after $P2$. These check bits result from a bitwise XOR operation between the symmetric segments, effectively expanding the control word length by $\frac{n}{2}$. Consequently, the matrix dimensions change from $m \times n$ to $m \times (n + \frac{n}{2} + 2)$. The $\frac{n}{2}$ increase arises from check bit inclusion, with an additional 2 bits introduced due to these check bits.

The decoding process for the proposed SYMSEG codes begins with segment-wise parity calculations in the control word. It involves three XOR operations: first, computing the expected control word segment by XORing the complementary segment of the encoded control word ($\widehat{CW}$) with the check bit (Cb); second, generating the syndrome ($S[n/2 \dots (n-1)]$) by XORing the expected and encoded segments to identify upset positions; finally, correcting identified upsets by XORing the syndrome vector with the current encoded control word. These steps ensure upset correction and detection of control word generation as shown in Figure 2.

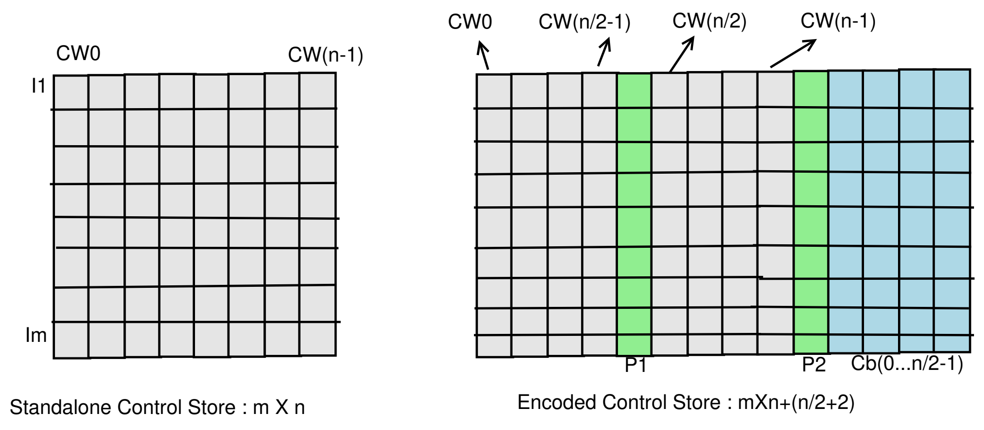
CW0
CW(n-1)
I1
Im
Standalone Control Store : m X n
CW0
CW(n/2-1)
CW(n/2)
CW(n-1)
P1
P2
Cb(0...n/2-1)
Encoded Control Store : mXn+(n/2+2)

Figure 1: Overview of Standalone and Encoded Control Store of MPCU Architecture

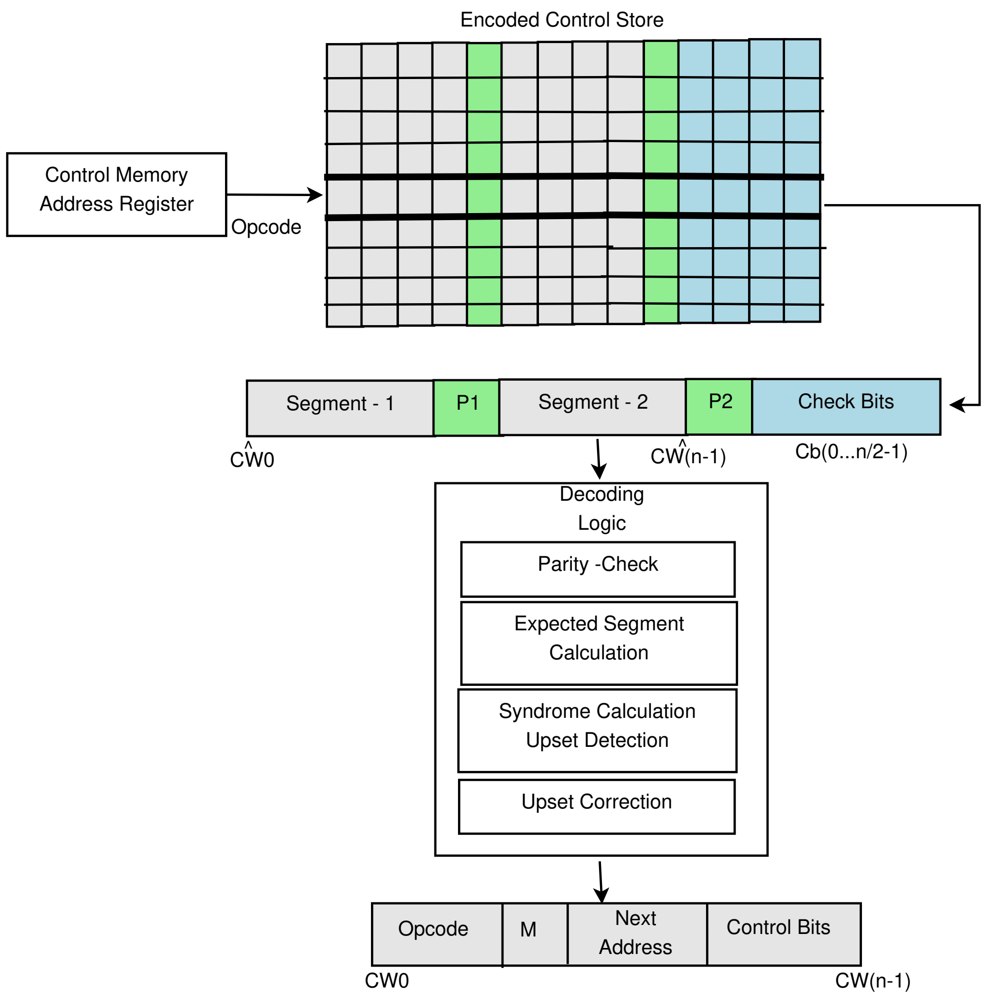

Figure 2: Design Flow of Decoder logic of proposed SYMSEG code

The efficiency of the proposed ECC hardware is evaluated using four metrics: *Code Rate, Resource Utilization, Fault Mitigation Efficiency, and Path Delay*. Regardless of the control word length, the SYMSEG bit codes consistently achieve uniform code rates, maintaining a fixed code rate of 0.66 by adding $\frac{n}{2}$ check bits and two parity bits to any control word of length n. Variability in correction bit counts, particularly in RS codes, is indicated by the 'upset*' value in Table 1. Results demonstrate the proposed ECC's enhanced fault correction ability compared to other codes, maintaining a steady code rate while improving error correction and detection capabilities. The proposed encoder and decoder logic incurs a 23% higher average area overhead than SECDED and Hsaio codes (as shown in Gracia-Moran *et al.* (2018)), yet it is 36% lower for BCH codes designed by Xie and Zhang (2021) and 18.67% lower for RS codes implemented by Kavian *et al.* (2005). Computing the longest syndrome bit primarily determines the decoding logic's path delay. Including the proposed ECC, the path delay measures 49.7 nsec, approximately 23 nsec greater than the standalone MPCU in A3PE3000 FPGA.

Code-Type	Code Length (# bits)	Check bit (# Bits)	Code Rate	Upset Detected	Upset Corrected
SEC-DED	16	5	0.761905	2	1
Hsaio Gracia-Moran *et al.* (2018)	16	6	0.727273	2	1
Dutta Zhu *et al.* (2011)	16	7	0.695652	2	2
BCH Xie and Zhang (2021)	16	10	0.615385	2	2
RS Kavian *et al.* (2005)	16	9	0.64	4*-6*	1*-3*
Proposed	16	8	0.666667	3	3*
SEC-DED	32	7	0.820513	2	1
Hsaio Gracia-Moran *et al.* (2018)	32	7	0.820513	2	1
Dutta Zhu *et al.* (2011)	32	7	0.820513	2	2
BCH Xie and Zhang (2021)	32	12	0.727273	2	2
RS Kavian *et al.* (2005)	32	12	0.727273	5*-8*	1*-4*
Proposed	32	16	0.666667	7*	7*
SEC-DED	64	8	0.888889	2	1
Hsaio Gracia-Moran *et al.* (2018)	64	8	0.888889	2	1
Dutta Zhu *et al.* (2011)	64	9	0.876712	2	2
BCH Xie and Zhang (2021)	64	14	0.820513	2	2
RS Kavian *et al.* (2005)	64	15	0.810127	5*-8*	1*-5*
Proposed	64	32	0.666667	15	15*

Table 1: Fault Mitigation Probability and Code Rate of proposed SYMSEG code in comparison to Related works

4.2 Fault Mitigation Approach in HWC architecture:

The HWC comprises a combinational circuit generating control signals and a memory element storing the operating state. Due to its sequential nature, traditional redundancy methods pose latency issues, as stated by Lohn *et al.* (2003); hence, a passive fault tolerance mechanism is needed. To avoid challenges caused by implementing redundancy techniques to hardwired circuits, our work integrates evolvable hardware, explicitly leveraging the self-healing capability to mitigate faults. We focus on reconfiguration through the FPGA-generic VRC technique. Despite VRC advantages, challenges like extensive architecture and non-partial reconfiguration in real-time scenarios prompted our research. Our contribution involves:

(a) Designing a compact VRC structure to minimize hardware usage and architectural footprint.

(b) Introducing constrained evolution for partial reconfiguration in VRC to localize the fault.

(c) Implementing intrinsic Genetic Algorithm modules to expedite circuit communication to the target platform, differing from extrinsic Evolvable Hardware approaches.

4.2.1 Proposed Design of Compact VRC Architecture

Understanding genotype and phenotype parameters is pivotal for optimizing hardware systems in evolvable hardware. The genotype space, $|G|$, manipulated by EA, encodes the target circuit with respect to select lines bit, which is supplied to the multiplexer of VRC architecture. The phenotype represents actual circuit behavior, evaluated by the EA for fitness. A compact VRC-based circuit will be exhibited only if redundant-free genotype space (2^N, where N is the length of the configuration register) aligns with a phenotype space expressed as O^I (where $I = 2^i$ and $O = 2^o$, i and o represents the number of input and output bits for circuits). In our proposed solution for designing compact VRC architecture, the Boolean function's sub-expressions eliminate redundancy in the configuration register, aligning the genotype and phenotype spaces closely. Detecting and relocating redundant logic outside the VRC reduces columns in the two-dimensional VRC architecture, reducing the search space for EA and enabling quicker convergence toward non-faulty solutions. This process ensures that the genotype space, determined by the configuration register length, aligns closely with the phenotype space, representing the circuit's Boolean expression. The resulting VRC circuit achieves a more compact and efficient representation by removing redundancy from two-dimensional space. The above updation is of utmost importance as it reduces the search space for EA and facilitates faster convergence toward non-faulty configuration.

4.2.2 Proposed Design of Partial Genotype Representation for Fault Localization and Constrained Evolution

In the VRC architecture for EHW, transmitting all N configuration bits from the register to the EA might lead to an expansive search space encompassing non-faulty configurations. Our proposed approach introduces a method similar to DPR to pinpoint potential fault locations by exhaustively injecting faults into the configuration register. This simulation has aided in identifying the pattern between the fault's location and the affected output bits. Based on this pattern, a frame encoder is designed to aid the VRC in localizing the faulty bits. The potential faulty bits in the configuration register begin at position $e \times n$, where e represents the position of the erroneous output bit, and n is the number of select line bits supplied to each programming element. The window expands to cover the successive $n-1$ positions for the respective PE. As a result, only the portion of $Config\_Reg$ from the $start\_index$ to the end_index is supplied to EA for evolution.

Within the VRC architecture comprising multiple (PE), we calculate the length of the configuration register (N) as $|PE| \times n$, considering each PE_i with n-bit select lines. In contrast to the standard approach where the entire $|PE| \times n$ is sent as input to the EA, our method selectively transmits $|pe| \times n$ bits, reducing the search space and focusing solely on the relevant faulty configuration bits. This focused approach is possible due to the influence of a subset of PEs ($pe \subseteq PE$) on specific output signals ($y_i \in Y$). By identifying the faulty output ($\bar{y}_i$), we confine potential fault locations to specific pe_i configuration bits. Thus, only the relevant $|pe| \times n$ bits are communicated to the EA, significantly reducing the search space. Leveraging the continuous placement of PE configuration bits in the register, our proposed algorithm localizes a configuration window delineated by start and end addresses.

4.2.3 Proposed Design of Intrinsic Genetic Unit

The genetic unit incorporates an EA designed for fault mitigation, deployed as a digital circuit on the FPGA. In our proposed design of the genetic unit, the chosen evolutionary approach is the genetic algorithm. The various components of genetic algorithms—such as Random Population generation, Fitness Calculation, Selection, Crossover, and Mutation are instantiated as digital circuits within the genetic unit module. Once a fault occurs, these modules communicate with the VRC module to transfer the genotype (Configuration content). The main contribution to the design of the genetic unit is the inclusion of a digital circuit responsible for the $FrameEncoder$, a crucial component facilitating constrained evolution. The interface between the VRC module and the ge-

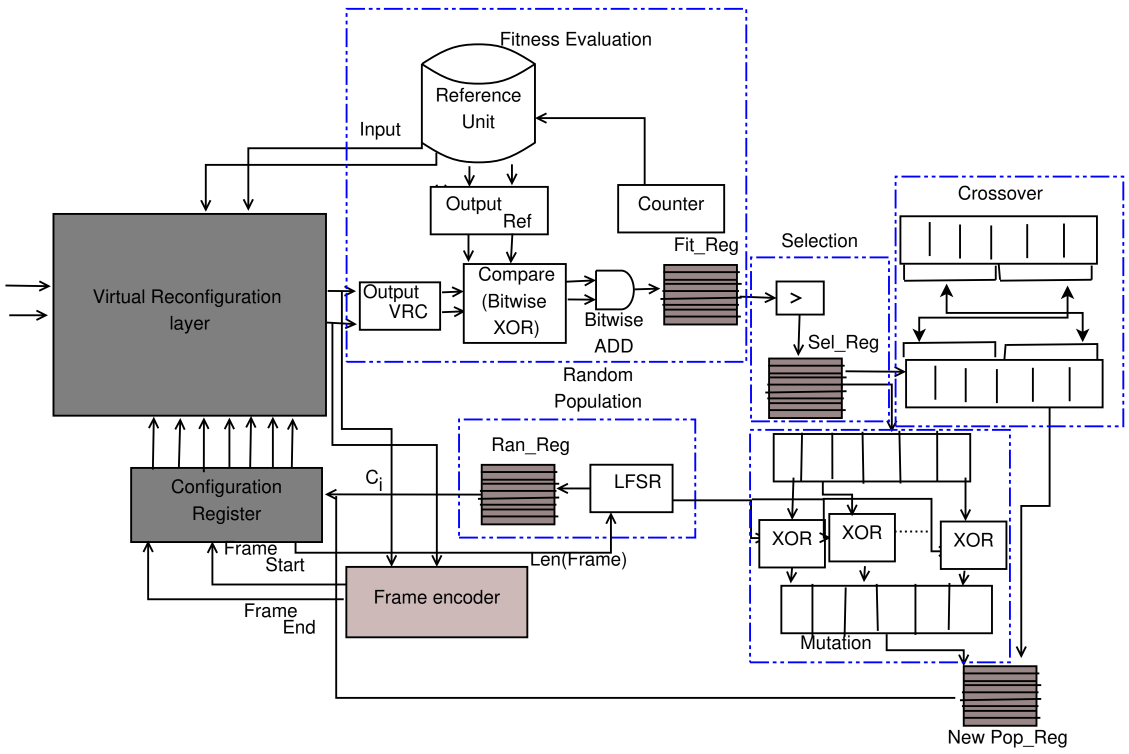

Figure 3: Proposed Genetic Unit: Interaction with Configuration Register and GA Modules- Random Population Generation, Fitness Evaluation, Selection, and Reproduction

netic units is shown in Figure. 3. The configuration register content acts as the genotype or chromosome of the genetic algorithm. Contrary to the standard genetic algorithm, only a part of a chromosome is evolved. The $FrameEncoder$ restricts the chromosome from the configuration register. The frame's start address, denoted as $Frame_{start}$, and end addresses as $Frame_{end}$ are decided based on the erroneous output bit position p and length of the select line of each programming element n. The value of p is calculated by comparing VRC's output, denoted as $Output_{VRC}$, and reference output, denoted as $Output_{Ref}$. The length of this frame, denoted as $Len(frame)$, is communicated as the primary input to the random population module.

The proposed fault mitigation approach is validated based on three key metrics: *1) Resource Utilization, 2) Fault Convergence, and 3) Fault Recovery Time*. The experiment is conducted on the control circuits used extensively on space-based digital systems found in autonomous vehicles, such as the control circuit of BLDC, the processor in RISC-V architecture, and other benchmark control circuits. It is evident from Table 3 that a reduction in most of the parameters influencing the VRC architecture is observed in our proposed VRC architecture after implementing the solution discussed in Section.4.2.1. In Table 2, the speedup achieved after including the solution discussed in Section 4.2.2 is summarized. From the above two observations, it is clearly evident that the proposed solution has improved the scalability and acceleration of the evolvable hardware-based approach in fault mitigation (both SEU and MBU). In addition to these results, resource utilization in terms of FPGA resources consumed and fault convergence in terms of the number of generations taken by the genetic unit is discussed in the thesis.

Table 2: Performance of proposed EHW system (Proposed FT) after *Constrained Evolution* in comparison to Standard FT adopted by Zhu *et al.* (2017) in terms of fault recovery time for the control circuit

Circuit	SEU		% of decrease in Fault Recovery time for Proposed FT	MBU		% of decrease in Fault recovery time for Proposed FT
	Standard FT	Proposed FT		Standard FT	Proposed FT	
BLDC	21.2	0.92	95.7	18.1	2.34	5.2
RISC-V	41.6	2.96	92.9	67.5	0.86	66.2
S1494	26.8	1.48	94.5	65.14	14.53	42.8
S510	36.4	2.44	93.3	220	123.45	43.9
S832	42.8	3.08	92.8	370	85.89	76.8
S420	23.5	4.12	82.5	33	29.65	10.2
S820	302.8	14.28	95.3	1250.4	189.9	84.8

4.3 Comprehensive Fault Mitigation: A Two-Layer Approach

The proposed EHW approach poses a significant challenge: components within the GA, such as fitness calculation and selection, encounter comparable probabilities of faults similar to those in the control circuit. This similarity arises because the GA unit and the control circuit reside in the same FPGA, encompassing numerous memory components (registers) for chromosome storage during evolution and essential logic functions for fitness evaluation and reproduction. This motivates our proposal of a two-tier bio-inspired mitigation technique: Tier 1 functions as the GA module to address faults in the control circuit. Tier 2 draws inspiration from immunotronics architecture to protect the GA modules, ensuring comprehensive self-repairing capabilities for control

Table 3: Reduction in VRC Array parameters in Proposed Compact VRC Architecture in comparison to standard VRC Architecture

Circuit	**Standard VRC Zhu *et al.* (2017)**				
	VRC Array Size	**# PE**	**# Multiplexer**	**Configuration size per PE**	**Genotype Length**
BLDC	6 × 3	18	54	8	153
RISC-V	7 × 78	546	1638	35	19110
S1494	5 × 52	780	780	26	6760
S510	6 × 11	198	198	21	1386
S832	8 × 56	1344	1344	32	14336
S420	7 × 4	84	84	12	336
S820	10 × 21	630	630	34	6300
Proposed VRC					
BLDC	6 × 1	6	12	6	36
RISC-V	7 × 35	245	735	15	3850
S1494	5 × 36	180	540	12	2160
S510	6 × 6	36	108	14	504
S832	8 × 26	208	624	18	3744
S420	7 × 2	14	42	7	98
S820	10 × 13	130	390	23	2990
	Reduction VRC Array size (#columns)	**Reduction in #PE**	**Reduction % in #Multiplexer**	**Reduction in Configuration size per PE**	**Reduction % in genotype length**
BLDC	0 × 2	12	77.8	2	76.47
RISC-V	0 × 43	301	55.1	20	79.85
S1494	0 × 16	600	30.8	14	68.05
S510	0 × 5	162	45.5	7	63.64
S832	0 × 30	1136	53.6	14	73.88
S420	0 × 2	70	50.5	5	70.83
S820	0 × 8	500	38.1	11	52.54

circuits. In the second layer of protection, immune cells are deployed for crucial stages in a genetic algorithm, such as fitness value calculation, selection, and reproduction. The stages of a genetic algorithm are variable in terms of their functionality; hence, a distributed and variable immunization layer is proposed.

In the proposed Immune cell-1, the reference unit is checked consistently for faults by generating antibodies for the target input and output bit-string. The antibody for each pair of input and output (I/O) strings is calculated using bitwise XOR and stored alongside. A parity bit is utilized to monitor the integrity of the antibody. Any discrepancy detected in this parity bit indicates a fault within the antibody, necessitating a re-computation of the antibody from the I/O pair. The second immune cell for the selection module identifies SEU in the information bit by performing the bit comparison in the fitness calculation generated chromosome and the chromosome in the selection module. The presence of one in the result indicates the occurrence of SEU, which will signal to scrub the selection register and re-enter the chromosome value. Immune Cell 3 prioritizes safeguarding the functionality of the mutation operator, assuming information level protection by Immune Cell 2. In this scenario, verifying functionality involves introducing a random zero or one bit to the mutation module, irrespective of the stage in the GA module. The resulting output undergoes verification against the expected output of a non-faulty NOT gate. If an unexpected value is obtained, the backward recovery process involves reconfiguring and applying the required configuration for the NOT gate. The FPGA-ProAsic 3e hosts the implementation of the control circuit, GA, and Immune cells. Compared to existing approaches, our EHW+Immune cell approach marginally increases resource utilization while ensuring higher reliability. Even after adding the second layer of protection, the fault recovery time is reduced by 88% on average compared to the standalone EHW solution, as shown in Table 4. Similarly, the fault mitigation efficiency increases by approximately 13% compared to the standard EHW procedure.

Table 4: Fault Recovery Time for Proposed Two-Tier Architecture

Control circuit	**Proposed Tier- 1 EHW (Intrinsic)**	**Proposed Tier -2 (Immunotronics)**	**Total Proposed Fault Recovery Time**	**Existing Tier-1 Zhu *et al.* (2017) (EHW Hybrid)**	**Speed Up**
BLDC	0.92	0.56	1.48	21.2	0.93
RISC_V	2.96	1.34	4.3	41.6	0.9
S1494	1.48	0.67	2.15	26.8	0.92
S510	2.44	1.21	3.65	36.4	0.9
S832	3.08	1.45	4.53	42.8	0.89
S820	4.12	2.67	6.79	23.5	0.71
S420	14.28	6.72	21	302.8	0.93

5 Conclusion

In this work, we have developed fault mitigation techniques to identify and rectify SEU and MBU in control circuit hardware. We devised a distinct fault mitigation solution through exhaustive fault injection and by considering the varying characteristics of the control circuit design. For the MPCU, we introduced an ECC-based technique leveraging the self-invertive property of the XOR gate. Our proposed encoder and decoder logic demonstrate an average area overhead of 23% higher than SECDED and Hsaio codes but 36% lower for BCH and 18.67% lower for RS codes. Furthermore, our approach showcases improved performance, achieving up to $(\frac{n}{2} - 1)$ upset correction with a consistent code rate of 0.66. Concerning path delay, our ECC code decoder shows six layers fewer than the BCH code and three layers more than RS codes for an 8-bit code length. In our fault tolerance solution for HWC based on EHW, logic resource consumption amounts to 56%. Conversely, the redundancy technique (TMR-Triple Modular Redundancy) incurs a 200% area overhead. Compared to related works, our approach reduces the use of multiplexers by 77%, thus cutting the genotype length by 63%. Our FT method achieves a 73.04% speedup in fault mitigation for the BLDC circuit compared to fault recovery times documented in related works. Augmenting our proposed VRC-based EHW approach with a third layer of protection inspired by immunotronics increases marginal resource utilization while significantly enhancing reliability and reducing fault recovery time. Even with the addition of the second layer of protection, fault recovery time is reduced by 88% compared to the standalone EHW solution. Similarly, fault mitigation efficiency increases by approximately 13% compared to the standard EHW procedure.

6 Organization of the Thesis

The proposed outline of the thesis is as follows:

(a) Chapter 1: Introduction.

(b) Chapter 2: Related works.

(c) Chapter 3: Fault Mitigation using Information Redundancy Technique for Microprogrammed Control Unit.

(d) Chapter 4: Fault Mitigation Using Evolvable Hardware Approach for Hardwired Control Circuit.

(e) Chapter 5: Fault Mitigation in Evolvable Hardware for Comprehensive Fault Mitigation.

(f) Chapter 6: Conclusion and Future Scope.

7 List of Publications

I. REFEREED JOURNALS BASED ON THE THESIS

1. Deepanjali, S Sk, Noor. *Self Healing Controllers to Mitigate SEU in the Control Path of FPGA Based System: A Complete Intrinsic Evolutionary Approach.* Journal of Electronic Testing. *Volume 38. Pages 1-19 (2022).*
2. Deepanjali, S. Mahammad, S. *A twofold bio-inspired system for mitigating SEUs in the controllers of digital system deployed on FPGA.* The Journal of Supercomputing. *1-33(2023).*DOI: 10.1007/s11227-023-05804-0.
3. Deepanjali.S, Noor Mahammad.Sk. *Scalable and Accelerated Self Healing Control Circuit using Evolvable Hardware*, ACM Transactions on Design Automation of Electronic Systems- Accepted on November 2023
   DOI: 10.1145/3634682
4. S Deepanjali and Noor Mahammad. *"A Novel ECC Scheme for Single and Multi-Bit upset Mitigation in Microprogrammed Control Unit"*, Submitted to Journal of Electronic Testing September 2023.

II. PRESENTATIONS/PUBLICATIONS IN CONFERENCES BASED ON THE THESIS

1. Chandrasekhar BS, Deepanjali S, Sk Noor Mahammad, "*Fault Tolerant Technique for Processor Control Path to Mitigate SEUs in FPGA"*, In 2022 IEEE International Symposium on Smart Electronic Systems (iSES), *pp. 31-35, Dec 2022.*
2. S Deepanjali and Noor Mahammad, *"Scalable and accelerated Evolvable Hardware"*, is presented in the **Student research forum track** in 2023 *36th International Conference on VLSI Design and 2023 22nd International Conference on Embedded Systems (VLSID),* Hyderabad, India.

3. S Deepanjali and Noor Mahammad, *“Design of Virtually Reconfigurable Hardwired Control Circuit Tolerant to Single Event Upset”* is Accepted in *37th International Conference on VLSI Design and 22nd International Conference on Embedded Systems (VLSID)*, Jan 2024.
4. S Deepanjali and Noor Mahammad, *“Immunotronics inspired Novel Self Repairing Finite State Machine for RISC-V based Processor”* in Proceedings of *National Conference on Systems Approach for Self-Reliance in Advanced Technologies (SASAT-2023)*, organized by DRDO.

References

1. **Gracia-Moran, J.**, **L. J. Saiz-Adalid**, **D. Gil-Tomas**, and **P. J. Gil-Vicente** (2018). Improving error correction codes for multiple-cell upsets in space applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **26**(10), 2132–2142.
2. **Hailes, P.**, **L. Xu**, **R. G. Maunder**, **B. M. Al-Hashimi**, and **L. Hanzo** (2015). A survey of fpga-based ldpc decoders. *IEEE Communications Surveys & Tutorials*, **18**(2), 1098–1122.
3. **Kavian, Y.**, **A. Falahati**, **A. Khayatzadeh**, and **M. Naderi** (2005). High speed reed-solomon decoder with pipeline architecture. *In Second IFIP International Conference on Wireless and Optical Communications Networks, 2005. WOCN 2005.*. IEEE.
4. **Kim, J.**, **N. Hardavellas**, **K. Mai**, **B. Falsafi**, and **J. Hoe** (2007). Multi-bit error tolerant caches using two-dimensional error coding. *In 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*. IEEE.
5. **Lohn, J.**, **G. Larchev**, and **R. DeMara** (2003). A genetic representation for evolutionary fault recovery in virtex fpgas. *In International Conference on Evolvable Systems*. Springer.
6. **Sekanina, L.** and **S. Friedl** (2004). On routine implementation of virtual evolvable devices using combo6. *In Proceedings. 2004 NASA/DoD Conference on Evolvable Hardware, 2004.*. IEEE.
7. **Silva, G. N. P.** and **R. O. Duarte** (2018). Towards evolvable hardware and genetic algorithm operators to fail safe systems achievement. *In 2018 IEEE 19th Latin-American Test Symposium (LATS)*. doi:10.1109/LATW.2018.8349669.
8. **Trimberger, S.**, **J. Moore**, and **W. Lu** (2011). Authenticated encryption for fpga bitstreams. *In Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*.
9. **Wang, J.** and **J. Liu** (2017). Fault-tolerant strategy for real-time system based on evolvable hardware. *Journal of Circuits, Systems and Computers*, **26**(07), 1750111.
10. **Xie, Z.** and **X. Zhang** (2021). Miscorrection mitigation for generalized integrated interleaved bch codes. *IEEE Communications Letters*, **25**(7), 2118–2122, doi:10.1109/LCOMM.2021.3074461.
11. **Zhu, M.**, **L. Y. Xiao**, **L. L. Song**, **Y. J. Zhang**, and **H. W. Luo** (2011). New mix codes for multiple bit upsets mitigation in fault-secure memories. *Microelectronics journal*, **42**(3), 553–561.

12. **Zhu, P.**, **R. Yao**, and **J. Du** (2017). Design of self-repairing control circuit for brushless dc motor based on evolvable hardware. *In 2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE.