Immunotronics Inspired Novel Self Repairing Finite State Machine for RISC-V based Processor

Deepanjali.S[1*], Noor Mahammad[1], Balasubramanian.P[2] and G Venkat Reddy[2]

[1]Indian Institute of Information Technology, Design, and Manufacturing, Kancheepuram,
[2]Research Centre Imarat, DRDO, Hyderabad

ABSTRACT

RISC-V processor is an open-source Instruction-set architecture (ISA), used to implement soft-core processors or System-on- Chip (SoC) in multiple FPGA and embedded systems. The increasing reduction in the size of these devices has increased the probability of upsets towards radiation environments. These errors are called Single Event Upset (SEU) which can cause functional failures in mission-critical components like control circuits. Hence reliability through self- healing mechanisms has become an essential characteristic of digital systems. Motivations from the human system to resist the invasion of harmful foreign bodies have been incorporated into the field of fault-tolerant electronics under the research area called Immunotronics. Our proposed system introduces self-repairing capabilities for the RISC-V control circuit through Immunotronics. The results are summarized as a proof-of- concept to verify the SEU-Tolerance.

Keywords: *Fault-Recovery, Immunization, Bio-Inspired fault-tolerance,SEUMitigation*

1. INTRODUCTION

The RISC-V Instruction SET architecture is used in multiple processors to execute 32/64-bit instruction in five stages as shown in Fig 1. The Instruction is fetched from the Instruction memory consisting of two main sections such as opcode, and operand address of registers containing the source value, and destination address. The Instruction decoding logic is the main control unit of the Processor whose functionality is described using a behavior model called Finite State Machine (FSM) as shown in fig 2.

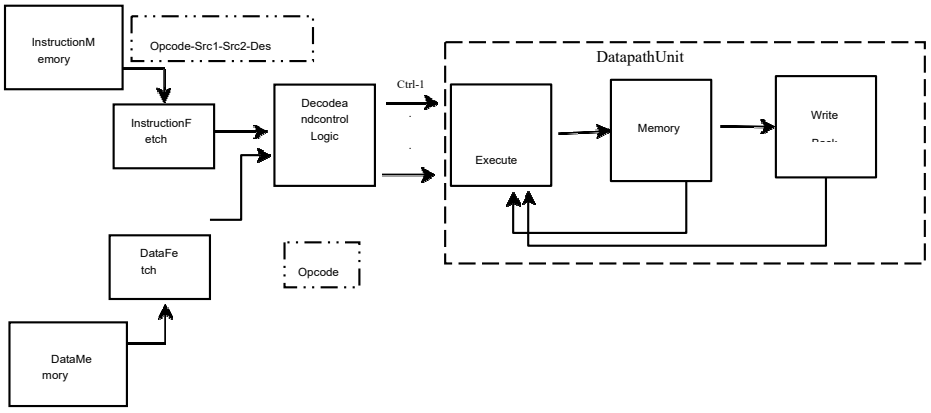

Figure 1: RISC-V Processor Design Phases

*Corresponding Author: E-mail: cs21d0001@iiitdm.ac.in,noor@iiitdm.ac.in

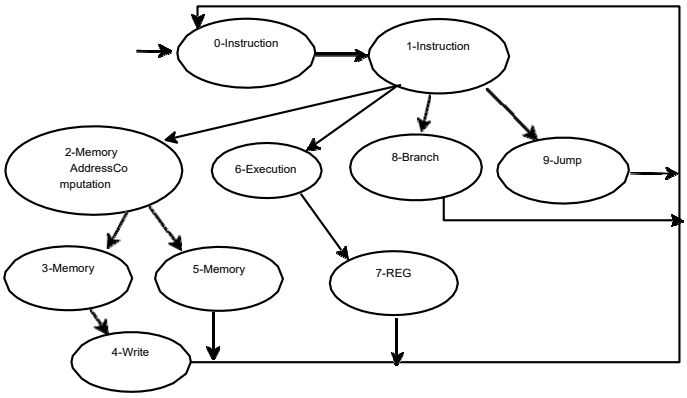

Figure 2: Behaviour Model: Finite State Machine of RISC-V Processor Control circuit
Immunotronic: RISC-V Processor

The FSM contains in total of ten states numbered from State 0-9 as shown in Table 2. Each input state on accepting the opcode from the decode unit transits to the output state. The details of the Opcode are given in table 1. At each outputstate, respective control signals such as ALUSRC A, PCWrite, RegA, MemRead, Memwrite, etc. are signaled as 1 and communicated to the Data path unit such as ALU and Memory for further execution. On occurrence of SEU in the FSM of the control logic functional error or unknown state transition can happen. For instance, Consider the Instruction Decode Stage where Stage 1 accepts the input opcode 0-000000 (Branch-if-equal) and reaches state 7-Branch completion. IF SEU happens in the second Bit from LSB then the operation related to state 9-Jump completion is performed releasing an erroneous control signal to the data path elements.

Table 1: Opcode and its corresponding Instruction

Opcode	OP0	OP1	OP2	OP3	OP4	OP5	Instruction
35	1	0	0	0	1	1	R-Type
43	1	0	1	0	1	1	j
0	0	0	0	0	0	0	Branch if Equal
4	0	0	0	1	0	0	Load Word
2	0	0	0	0	1	0	Store Word

Table 2: FSM Table of RISC-V Processor Control Circuit

States	Input State	Opcode						Output state
		OP0	OP1	OP2	OP3	OP4	OP5	
0	0000	x	x	x	x	x	x	0001
1	0001	1	0	0	0	1	1	0010
1	0001	1	0	1	0	1	1	0010
1	0001	0	0	0	0	0	0	0110
1	0001	0	0	0	1	0	0	1000
1	0001	0	0	0	0	1	0	1001
2	0010	1	0	1	0	1	1	0011
2	0010	x	x	x	x	x	x	0101
3	0011	x	x	x	x	x	x	0100
4	0100	x	x	x	x	x	x	0000
5	0101	x	x	x	x	x	x	0000
6	0110	x	x	x	x	x	x	0111
7	0111	x	x	x	x	x	x	0000
8	1000	x	x	x	x	x	x	0000
9	1001	x	x	x	x	x	x	0000

ISBN - 978-93-95038-50-8

Hence self-healing mechanisms are required for Decoding and Control logic shown in Fig 1. The traditional approach chosen for dealing with the SEU tolerance is generally Hardware redundancy or Information redundancy where the control logic is replicated three times with Voting logic or FSM state encodings are added with redundant data encoding respectively. In both these cases area and power increase as the redundancy rate increases. In a lightweight and cost-effective digital system redundancy is not appreciable. Hence a bio- inspired system-level self-repairing concept motivated by a complex Immune system found in organisms is adopted in our proposed system.

2. BIO-INSPIRED FAULT TOLERANCE

The Bio-Inspired Fault tolerant system is a concurrent, online system that per- forms Fault detection, minimization, and recovery procedures. For the past four decades, this field has been used in multiple STEM domains. But only for a few years, the implementation of these concepts are incorporated in real-time systems. In general bio-inspired system mimics the adaptability, Learning, and self-organizing nature found in different organisms. The living organism's tendency to sense and adapt to nature is the core concept that motivates bio-inspired systems [10]. These systems are influenced by the evolving nature of organisms to produce a better population in different phases of life is mimicked in electronics called as Evolvable Hardware [7,8,9]. The organism's multicellular development from a primary cell and the nature of cell replacement is adopted in Embryonic [4,5,6]. Similarly, the learning characteristics of the organism to fight and survive harmful foreign invasions are studied under Immunotronics. In general, the multilayered nature such as Physical, Psychological, innate, and acquired protection such as skin, acidity, Macrophages, and Cell-mediated immunity found in living organ- isms is adapted to electronics such as Physical enclosure, Temperature control, Triple modular redundancy, and Immunotronics respectively.

In our proposal, the fourth area of bio-inspired electronics is explored and implemented. The immune cells (T-cells) in the human system possess a memory that is used to detect harmful/Non-harmful organisms. This acquired immunity is mapped to FSM correct/incorrect transition in the error-detection process. In the memory of the human immune system called as T-cell the gene are formed together to create Antibodies. Similarly, for Fault tolerance electronics, the Error Recovery component creates admissible conditions that are required for the appropriate functioning of the electronic circuit. The gestation learning performed by T-cells is achieved in FSM fault tolerance during the training stage and clonal detection is mapped to negative selection-based recognition of acceptable conditions in fault-tolerant electronics.

3. RELATED WORKS

The works of [1] have introduced the immunological aspect of hardware fault tolerance. The introductory implementation proceeded with XCV300 FPGA. The FSM of the basic 0-10 counter circuit is designed along with immune cell modules. The proposed system is verified for stuck-at-Faults by holding the state at its current value. The latched value of the states is set to either 0 or 1 for stuck-at-fault effects and a comparison value is provided with an Ok signal if the match is found. The authors of this work have considered the self/non-self-differentiation of faulty states for future work. The future work of the previous paper was continued in [2]. The noteworthy exception as mentioned by the author is to avoid distributed nature of FSM fault tolerance. The proposed work achieved a continuous and autonomous

implementation of fault tolerance in the same counter circuit with self/non-self-differentiation. The generation of tolerance conditions is one of the major requirements for the Immunization cycle. In this work, [3] a symbiotic evolution is used to generate the tolerance condition or otherwise termed antibodies. The conditions are generated usingsymbiotic evolution for decade counter and MCNC benchmark FSM circuits and compared with standard genetic algorithm generation.

4. PROPOSED DESIGN PHASES OF FSM IMMUNIZATION

The design phases for absolute fault Detection and recovery are illustrated for RISC-V processor, Although the design phases are generic to any FSM model of the hardware circuit. The hardware circuit represented in behaviour model of FSM is considered for Logic-Under-test for Immunization. The FSMs in general indicate the admissible state transition as shown in Fig 2. During any possible in- put given to the circuit, the FSM is considered reliable when states transit to only allowed output state represented in Table 2. The transitions from any individual state to another than an admissible state are considered as faulty operation of the control circuit. For RISC-V control circuit the fundamental function includes accepting the opcode and input state and transiting to output state. With respect to state transition table 2 and Table 1 there are 10 states each representing the operation performed on instruction based on the opcode of the instruction received. Hence the antibodies for RISC-V FSM are constructed using INPUT STATE / OPCODE / Output STATE / CONTROL Signals. For Instance, 0000/xxxxxx/0001/0000010010 represents that 0000 (Input State- Instruction Fetch) on receiving opcode of any combination 0's and 1's reaches 0001 (Output State-Instruction Decode) and signals control signal Decode, InsMemRead and DataMemRead. The antibodies are constituted by generating all combinations of input state and opcode for an FSM. Hence for RISC V processor consists of fifteen admissible antibody strings each of length 4 + 6 + 4 + 10 = 24 bits. The proposed design stage for the Immunization process includes 1) Antibodies Acquisition and Generation 2) On-Line Fault Detection 3) Fault Elimination as shown in Fig 3.

4.1 ANTIBODIES ACQUISITION AND GENERATION

The objective of antibody acquisition is to create a corpus of all possible functionalities or transitions admissible for the FSM. The corpus can be generated in two ways 1) Predetermined input states available in the State transition diagram or table 2) Applying random inputs to FSM. The latter approach is similar to the human immune systematic way of coping with nature through the learning process. In both approaches fault tolerant FSM is considered for the acquisition and generation. In our proposed design corpus is created in both approaches and compared. In general, the former approach is widely used due to its simplicity, only for circuits whose admissible transition is not available, the second approach is adopted. In general, second approach is also suitable since antibodies of 200,000 count can be generated in 10Sec which is considered the training stage for the Immune cells which is performed offline prior to operation.

The negative Selection algorithm used for intrusion detection in networks is used in our proposed design to generate strings G that fail to match antibodies in at least p positions. The other bio-inspired fault tolerance system such as Evolvable hardware and embryonic verifies the presence of a valid condition of fault-free operation. On contrary, the negative selection algorithm checks the strings generated in a random corpus with the antibodies. The strings

dissimilar at p positions are removed. Hence this negative selection process is generally united with the random acquisition of corpus.

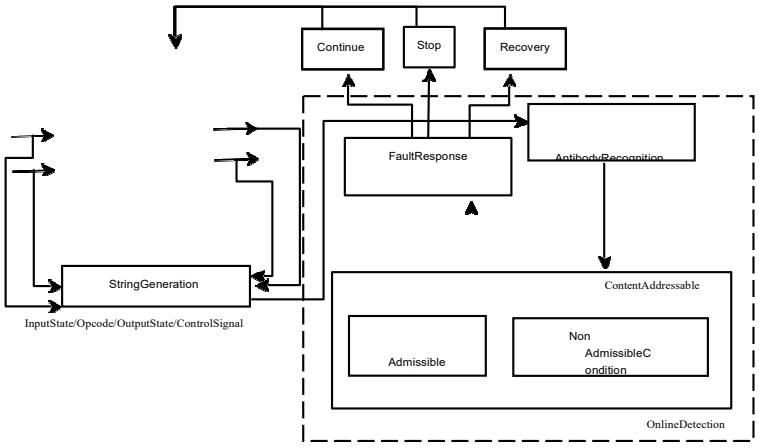

Figure 3: Block Diagram of Proposed Immunization design phases

4.2 ON-LINE FAULT DETECTION

The hardware immune system will be integrated with the FSM and perform as an interface to the hardware control logic. The detection mechanism includes main components such as the Control circuit module for fault repair - FSM of the circuit, Fault injection module which randomly injects the error in the opcode of the FSM, Under Non-Injection instance, the FSM produces a string with valid bits of input state, opcode, output state, and control signal. On fault injection, the strings dissimilar to the antibodies are generated by the FSM. A string initiation components generate the string in a format similar to the standard corpus discussed in the previous section, which is compared with the antibodies stored in the Immune cell memory. A content-addressable memory is used to store the antibodies. The indexed memory allows parallel searching and comparison for strings with the antibodies. The match is checked for p adjoining position. The comparison operation signals 1/0 representing the presence and absence of antibody match with the index of the antibody. The immune cell memory is 64-bit in length where 24 bits are used for data and the rest is used for address mask as per the requirement. The mask size and data bits are reconfigurable by the designer. The comparison is performed by window- based search. The window size is increased or decreased to ensure the found signal generation within the transition cycle of the FSM. For RISC-V detection of antibodies, 10 bits were set as the window size. The partial matching with the window method is similar human T-Cell mediated immune system. Hence the online detection of fault is performed for an FSM at every state instance. The generated string is compared with the antibodies for at least p- positions. The partial mapping allows detection to be made before the erroneous transition is made by the FSM, thereby prohibiting the wrong control signals.

4.2 FAULT ELIMINATION

The final stage of the proposed design cycle follows the fault removal stage. In many related works, complete immune system-based fault recovery solutions are not explored. In our proposed solution fault removal techniques are also motivated by the human immune

system. In the human immune system, the classic approach to be protected from outside invasion is achieved by fighting foreign bodies and neutralizing the ruinous behavior of these bodies. In the hardware system to neutralize the fault N-modular redundancy is followed. In accordance to FSM, the states can be triplicated with a voting logic. Hence the outputs from individual states are compared in the voting logic to choose the majority output, thereby faults can be tolerated when a match signal is not found. The faulty state identified by the voting logic is replaced with non-faulty in consecutive cycles. On contrary, the complete Immune-based solution would require a fault recovery procedure also to be placed in immune cell memory. For this recovery, states are also embedded in the antibodies stored in the immune cell. On the occurrence of a non-match signal, the recovery state of the current input state is performed. It is achieved in hardware tolerance by storing the recovery states in the multiplexer as a spare state. In further understanding, the anti- bodies attach with the antigen to protect from attacking normal cells. In FSM similar approach is obtained by maintaining a spare state, which requires precedent knowledge of the system operations. A complete bio-inspire fault tolerance can also be obtained by combining two other approaches such as embryonic or Evolvable hardware for fault elimination.

5. Results and Discussion

As a proof of concept the RISC-V Processor's Finite state machine is implemented in the Verilog on Zynq 7000 FPGA. The Immunotronics model as represented in Fig.3 along with fault injection module is deployed in the FPGA to verify the fault recovery model. The fault injection model consists of behaviour level implementation of XOR gate to flip the bits in the input of the Finite State machine and the state encoding. The Random generator module is designed to select the input and state encoding at random cycle. In the case of RISC-V processor the opcode is the input and states 0-9 are the state encoding. When the bits of the above two fault location are selected, it is applied to XOR gate with one of its input masked to Vcc. This process of bit flip causes an erroneous transition which has to be mitigated by Immunotronics module. The proposed design is compared traditional redundancy method called Triple Modular Redundancy and Hamming distance, where each state is triplicated along with a majority voter state, to select at most correct outputs from the three states and state encoding is attached with redundant bits respectively. The error recovery rate is high in proposed system since complete error recovery is possible with no redundant bits added to the state encoding; When compared to TMR the number of core cells are less but the Flash ROM usage to store the antibodies is high.

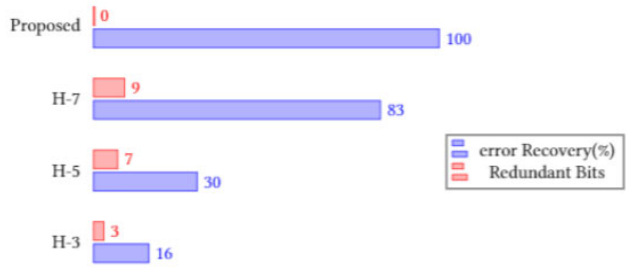

Figure 4: Comparison of error recovery in Information redundancy and Immunotronics in terms of Redundant Bits in RISC-V Processor

Table 3: Resource Utilization for TMR vs Proposed Design

	Proposed Design	TMR
Core Cell	122	145
I/O Cell	24	28
FlashROM	5Kb	0
Delay	17ms	0

6. CONCLUSION

The Fault tolerance approaches has increased requirement in designing reliable control circuit. In comparison to Redundancy based mitigation techniques the Immunotronics techniques has higher error coverage and less area utilization. The memory occupancy of Immunotronics plays huge role in hindrance of the development and real-time usage. Hence an alternative method to deploy the antibodies will be explored in our future works.

7. ACKNOWLEDGEMENTS

The author acknowledges the encouragement and support provided by Sh U Raja Babu, Outstanding Scientist & Director RCI and his team to continue and pursue the research activity.

REFERENCES

1. "Immunotronics: Hardware fault tolerance inspired by the immune system". In: Evolvable Systems: From Biology to Hardware: Third International Conference, ICES 2000 Edinburgh, Scotland, UK, April 17–19, 2000 Proceedings 3. Springer. 2000, pp. 11–20.
2. "Immunotronics-novel finite-state-machine architectures with built- in self-test using self-nonself differentiation". In: IEEE Transactions on Evolutionary Computation 6.3 (2002), pp. 227–238.
3. Sanghyung Lee et al. "A new immunotronic approach to hardware fault detection using symbiotic evolution". In: Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach: First International Work-Conference on the Interplay Between Natu- ral and Artificial Computation, IWINAC 2005, Las Palmas, Canary Islands, Spain, June 15-18, 2005, Proceedings, Part II 1. Springer. 2005, pp. 133–142.
4. Tyrrell, A., 2004. Embryonics and Immunotronics: Biologically Inspired Computer Science Systems. Computation in Cells and Tissues: Perspectives and Tools of Thought, pp.93-116.
5. Mange, D., Sipper, M., Stauffer, A. and Tempesti, G., 2000. Toward robust integrated circuits: The embryonics approach. Proceedings of the IEEE, 88(4), pp.516-543.
6. Mange, D., Sanchez, E., Stauffer, A., Tempesti, G., Marchal, P. and Piguet, C., 1998. Embryonics: A new methodology for designing field-programmable gate arrays with self-repair and self-replicating properties. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 6(3), pp.387-399.
7. Kalganova, T., 2000, July. Bidirectional incremental evolution in extrinsic evolvable hardware. In Proceedings. The Second NASA/DoD Workshop on Evolvable Hardware (pp. 65-74). IEEE.
8. Greensted, A.J. and Tyrrell, A.M., 2007. Extrinsic evolvable hardware on the RISA architecture. In Evolvable Systems: From Biology to Hardware: 7th International Conference, ICES 2007, Wuhan, China, September 21-23, 2007 Proceedings 7 (pp. 244-255). Springer Berlin Heidelberg.
9. Haddow, P.C. and Tyrrell, A.M., 2011. Challenges of evolvable hardware: past, present and the path to a promising future. Genetic Programming and Evolvable Machines, 12, pp.183-215.
10. Moreno, J.M., Thoma, Y. and Sanchez, E., 2006, June. Poetic: A Hardware Prototyping Platform With Bioinspired Capabilities. In Proceedings of the International Conference Mixed Design of Integrated Circuits and System, 2006. MIXDES 2006. (pp. 363-368). IEEE.