Novel Embryonics Adder Architecture with Unicellular Self-Check Unit

Sakali Raghavendra Kumar[1], S K Noor Mahammad[1(✉)], P. Balasubramanian[2], and G. Venkat Reddy[2]

[1] Design, and Manufacturing, Indian Institute of Information Technology, Kancheepuram, Chennai, India
noor@iiitdm.ac.in
[2] RCI, Hyderabad, Telangana, India

Abstract. FPGAs are prominent electronic devices used in mission-critical applications due to their reconfiguration mechanism. The mission-critical applications will be used in harsh radiation environments. Electronic devices are susceptible to single-event upsets because of radiation effects. Prior works used various traditional fault-tolerant mechanisms like redundancy and hardening techniques to overcome the SEUs. Those were not adaptive in nature. So, some researchers have moved towards the adaptive nature of fault-tolerant mechanisms. Embryonics is one of the branches of bio-inspired techniques that has an adaptive nature. This technique suits the implementation of fault-tolerant combinational circuits using cloning mechanisms like biological embryos. But it has limitations, like hardware redundancy, and the issue of an insufficiency of resources in FPGA. Our proposed approach would avoid redundancy and reduce the hardware overhead. It has been implemented on the adder circuit to determine its efficiency.

Keywords: Bio-inspired · Embryonics · Fault-Tolerant · FPGA · SEU · Adder

1 Introduction

Electronic systems are essential in industrial, scientific, medical, military, and daily life appliances. These should be reliable and available at any time to achieve user comfort and flexibility. But electronics systems are sensitive to radiation effects and high voltage spikes. These have an impact on soft errors in electronic systems and weaken the entire system as a result of a single problem, particularly in harsh environments such as space, submarines, and nuclear areas. To meet this challenge, devise a fault-tolerant system that is capable of healing the fault. Traditional fault-tolerant systems such as TMR (Triple Modular Redundancy) are used in most of the applications. Though TMR is reliable to achieve soft errors, it requires a lot of resources and is not self-adaptable. Bio-inspired systems can have a self-adaptable nature. These systems are inspired by nature, and this idea is most suitable in circuits to achieve self-healing, self-replication, and adaptability. The electronic tissue is called the PEOtic axis [1], derived from three axes of bio-inspired nature: phylogenesis, ontogenesis, and epigenesis. PEOtic has been

B. Raj et al. (Eds.): ICETE 2023, AER 223, pp. 928–935, 2023.
https://doi.org/10.2991/978-94-6463-252-1_93

used in developing fault-tolerant bio-inspired hardware based on its nature and usability. The phylogeny is related to the evolution of the species. The same concept is used in hardware to evolve the circuit using evolutionary algorithms. The epigenetic process is used to develop the individual through the learning process; this mechanism is used in immunotronics. Ontogeny is a cellular growth concept. In this case, the damaged cells will be self-repaired. The embryonics architecture is derived from the ontogenetic axis.

Embryonics is a combination of embryology and electronics [2]. The embryonics system's self-repair mechanism rests on hardware redundancy. It repairs on a fine-grained scale in lieu of the coarse-grained scale that was used in conventional fault-tolerant systems. In embryonics development, a failure cell can be replaced with a spare cell. The embryonics method was used in the majority of the work to develop fault-tolerant FPGA devices. FPGAs have reconfiguration capability and a common fabric surface to implement the desired circuits. Due to this, FPGAs are the best option for designing a fault-tolerant circuit using either embryonics or evolvable hardware. In our proposal, we design a fault-tolerant combinational circuit using an embryonics approach. This design will be independent of any kind of platform.

2 Literature Survey

Many attempts have been made to design an efficient fault-tolerant system that uses both embryonics and evolvable hardware in their own unique ways. We have gone through various existing works to know their designs and implementations. The authors [3] developed a self-checking based electronic with dual modular redundancy, a sequence test pattern for detecting faults in MUX-based logic modules, and a configuration bitstream register respectively. A comparison-based technique was used in embryonics electronics [4]. Antibody cells are used in the hardware in addition to the embryonics array. Here, the data of embryonics cells and the data stored in antibody cells were compared to detect faults. The authors [5] designed an embryonics system with dual modular redundancy and an XOR gate to detect the faults in the logic module and configuration memory. The hamming code is an information-based technique for detecting errors in memory [6]. The authors had used extended hamming code in their work to detect the SEUs in configuration memory [7]. In these works, self-repairing mechanism is a drawback which increases the hardware cost and reduces the computational performance.

An artificial immune system (AIS) was implemented on an FPGA using an embryonics array structure [8]. This is a control system that is used to direct the Khepera Robot in critical and complex environments. The AIS was used to detect faults during the working process of the robot. Gayatri et al. [9] designed a multiplier and adder circuits using embryonics architecture. Cloning was used to exploit configuration bit redundancy, and the circuit was evolved using a genetic algorithm to cover faults. In similar to this syncronous counter design is implemented with the sequential embryonics architecture [10]. Zhai Zhang et al. [11] introduced a FCRSS (Fault-Cell Reutilization Self-healing Strategy) technique to avoid transient faults with faulty cell reutilization. This method is demonstrated with a 4-bit adder. It had a process of elimination and reconfiguration. In the future, the fault cell will be transparent and reused. The intelligent fault-prediction and self-healing embryonics hardware is presented by Khaseem Khalil et al. [12]. In this

hardware, fault-prediction was done in each cell using a time delay. For every four cells in the embryonics array, a spare cell with the same functionality was used to replace the faulty cell. This required a more hardware resources for implementation. Existing works faced challenges such as complex cell structure, mapping, and self-repairing design in order to build an embryonic architecture with less hardware overhead.

3 Background Work of Embryonics System

As we discussed in Sect. 1, embryonics system mimics the biological embryo. Organs grow gradually with various functions through division and differentiation in the biological embryo. The other organs continuously produce necrosis cells in the process of operation, but the organ maintains normal function because of stem cells. The stem cells differentiate into the various functional cells to replace the dead cells, hence maintaining normal operation of the organ. The same mechanism is used in the embryonics hardware. In this hardware working and spare cells are arranged in an array structure. When the work cell gets damaged, a spare cell will be replaced with the faulty one, similar to biological stem cells. This makes functions available to the whole system at any time.

The embryonics structure will contain functional cells. These cells are arranged in a two-dimensional array structure. As shown in Fig. 1, spare cells are arranged in an array structure with functional cells to recover faults structure. In the existing embryonics cell, it contains the address generator, gene module, function module, self-repair module, and control module. The address generator module is used to coordinate with other cells in an array using the gene information determined by its location. Gene modules are used to store the configuration information of the embryonics cells, which mimic the DNA of biological cells.

The function module contains the logical circuit to perform the required operation based on configuration data. The control module contains the functionality to control the cell's working, fault, transparent, or idle state based on the signals. The input/output module is responsible for realising the input and output signals in order to communicate with other cells. The self-repairing module is used to detect the cell state and determine the working condition of the cell.

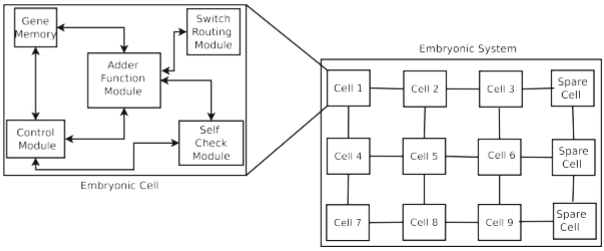

Fig. 1. Existing Embryonic Adder Cell

4 Proposed Work

In previous studies, each embryonics cell had a self-repair unit with redundancy techniques and gene memory. These are the primary reasons for rising embryonic hardware costs. We were inspired to create an embryonics architecture with a shared self-repairing unit, configuration memory unit, and control unit. The embryonic hardware is designed for an adder circuit in this work.

4.1 Embryonics Adder Cell

In this work, a fault-tolerant 8-bit adder has been designed using an embryonics architecture. The adder circuit is divided into two parts: the sum part and the carry part. The adder circuit can be designed in various ways. So we designed the functionality of the adder circuit using two 4:1 MUXes: one for sum and another for carry. In a similar way, two 8:1 MUXes are used to select the input bits from the input1 and input2 registers of the adder. Three selection bits are required to select the inputs from these MUXes for each bit addition. The selection bits will vary depending on the position of the adder. The 8-bit adder requires a total of 26 bits to select inputs. The output of input-1 MUX and input-2 MUX is used as selection bits for generating the result from the adder MUXes. Carry MUX in this adder will have two data bits, "0" and "1". These bits are also stored in the configuration register along with 24-bits that are used for selecting the input bits. This complete operation is framed as a single embryonics adder cell, as shown in Fig. 2. Later, the sum and carry bit will be passed to the self-checking unit through the control module.

4.2 Control Unit of Embryonics Cell

The control unit is one of the essential modules in the embryonics system. The embryonics adder cell can be activated as a working or dead cell by the control unit. The control unit selects the resultant bits from the current functional adder cell to transfer to the self-checking unit. The control unit will serve as an intermediary for adder cells and the self-repairing unit. This will determine the status of the adder cell after receiving the output from the self-checking unit. The control unit is also acting as a routing circuit to manage the neighbouring cells.

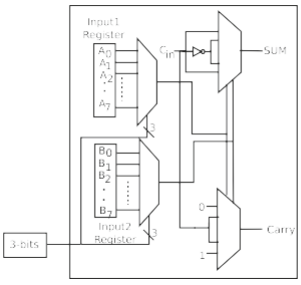

Fig. 2. Embryonic Adder Cell

4.3 Self-Checking Unit

The self-repairing unit contains the golden output generator, OR gate and two XOR gates. The golden output generator contains LUT modules that are used to store the full adder's sum and carry bits. The output bits from LUT modules will be selected with the input bits of the adder. As we have an 8-bit adder, the golden output generator will be operated eight times with the current input bits of the adder. In this, two LUT modules were utilised to store sum and carry bits respectively. The sum and carry bits of embryonics adder cell will be compared with output bits generated from the golden output generator. In the present work, we use the TMR technique on the golden output generator to avoid SEUs. We have used a comparison-based mechanism for detection. In this, XOR gate is used for comparison. Two XOR gates were used, one for the sum bits and another for the carry bits. These XOR gates will detect errors in the sum bits and the carry bits. The resulting bit from the XOR gates is '1', then it notifies that fault has occurred in the result. The OR gate is used to confirm the occurrence of the fault. If both outcomes are 0 bits, it sends "0" to the control unit; otherwise, it sends "1" to the control unit. Later, the result will be passed to the control unit. In the present work, we use the TMR technique on the self-checking unit to avoid SEUs.

5 Implementation of Novel Embryonics Adder

The embryonics adder design is inspired by virtual reconfigurable architecture. MUX was used to design modules of adder and self-check unit. The embryonics adder contains a group of adder cells, a control unit, configuration memory, and a self-checking unit. In the proposed work, we comprised the embryonics adder architecture with minimal hardware and tried to improvise the performance and efficiency. The adder cells are grouped together in the NXM matrix format. In this architecture, two spare adder cells were used along with working cells to avoid SEUs in a critical condition. This architecture will behave like a biological embryo. The spare cells have been cloned with the same functionality as the working cells. This spare cell will replace the faulty cell in the embryonics adder. The spare cells will remain idle if there is no error in the adder. In previous works, the self-checking unit and other modules like the gene module, control module, and coordination module were designed along with each functional module as shown in Fig. 1. This redundancy implementation is the primary reason for increasing hardware cost. So, the proposed work avoids redundant modules in the architecture. Using a common configuration register, control unit, and self-repair unit for adder cells as shown in Fig. 3.

Figure 3 illustrates the embryonics 8-bit adder design. Eight adder cells and two spare cells are arranged in a 2×5 matrix. These cells are interconnected with each other for communication with neighbouring cells. The control unit has controlling privileges over adder cells, configuration memory, and the self-check unit. The control unit has a counter and a three-bit register. The adder bit length will be assigned to the counter. The three-bit register will retrieve the information of each adder's MUX selection bits by using the counter condition. Based on the error signal provided by the self-check unit to the controller, the counter will increase the count value to pass the next adder's selection bits. Next the current adder operation will acknowledge the control unit with its result

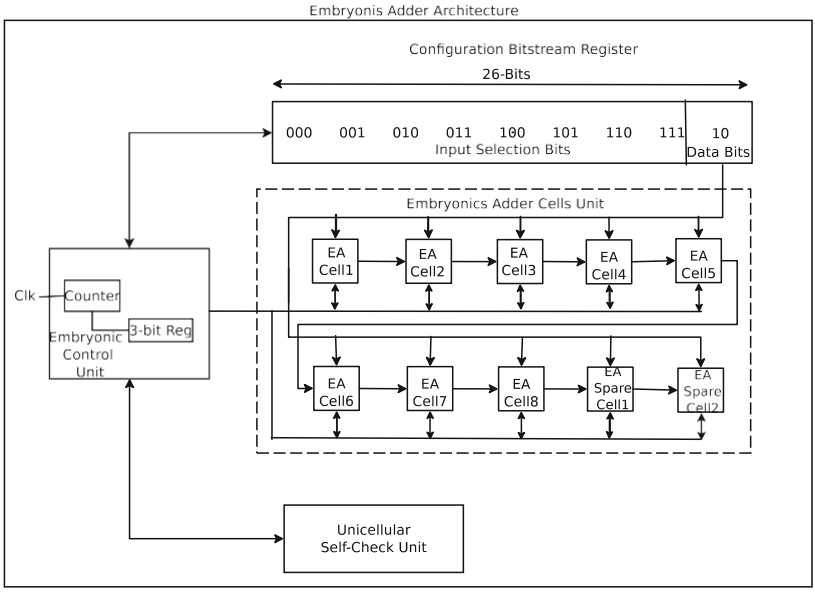

Fig. 3. Novel Embryonic Adder Architecture

for fault-checking. The controller will pass adder's sum and carry bits to the self-check unit to validate the result. The self-check unit will validate both the sum and carry bits. Either in the sum bit or in the carry bit causes a fault then self-check unit forwards the result as "1" to control unit. This result will be notified as an error signal by control unit. Then the current cell will be made into a dead cell, and the current cell's input selection bits will be forwarded to the next adder cell, and the addition operation will be repeated with the current cell inputs. If the self-checking unit's output is "0." It is indicating that there is no error signal and forwarding the carry bit to the next adder and also storing the sum bit in the output register. Instead of column and row elimination, fault cells are eliminated using a cell-forward approach in this work. As a result, we chose two spare cells for an eight-bit adder.

6 Results and Discussion

The proposed work's reliability was tested by a single-bit flip on the function selection MUX. It was tested at various function selection locations randomly for 30 iterations. A single-bit error was injected in the adder at each iteration. In these 30 faults, all were corrected successfully. The self-check and control units were successful in recovering faults with 100% error recovery rate. The proposed work's hardware costs 67% less than the standard embryonic adder, shown in Table 1. It works better than the existing work at recovering the faults in less time. It is simulated on the Xilinx Vivado IDE and targets the Zynq 7000 FPGA.

In the state of the art, the focus was on the self-checking unit instead of looking at embryonic cell architecture to address the hardware redundancy. This is a drawback

Table 1. Comparison of Hardware utilization

Works	Adder size	LUTs	IO Ports	Registers
Standard Embryonic Adder	8-bit	80	180	60
Proposed work	8-bit	26	54	6

in the embryonic system, as the embryonic cell has five modules, and the same cell has been duplicated N times. Then the redundancies will pile up, which will increase the hardware overhead. But the proposed work has only an adder unit along with input selection MUXes. Instead of using the same self-check unit in all embryonic cells, we have used a single self-check unit, which is commonly used by all adders. Here, configuration memory and the control unit were also used as a single unit for all adder cells. However, the main unit for fault tolerance is a self-check unit. So, the proposed research is known as embryonics adder architecture with a unicellular self-check unit.

7 Conclusion

The proposed work reduces the hardware overhead by eliminating the redundancy in embroyincs cells with a one-cell self-check unit. It works reliably and efficiently with an 8-bit adder. The proposed work will be implemented on the multiplier and large-bit adder to check performance in recovering faults. Also, we will plan for an information-based fault-tolerant approach for the configuration memory unit in our future work.

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. Andy M Tyrrell, Eduardo Sanchez, Dario Floreano, Gianluca Tempesti, Daniel Mange, Juan-Manuel Moreno, Jay Rosenberg, and Alessandro EP Villa. Poetic tissue: An integrated architecture for bio-inspired hardware. In *Evolvable Systems: From Biology to Hardware: 5th International Conference, ICES 2003 Trondheim, Norway, March 17–20, 2003 Proceedings 5*, pages 129–140. Springer, 2003.
2. Richard Canham and Andy Tyrrell. An embryonic array with improved efficiency and fault tolerance. In *NASA/DoD Conference on Evolvable Hardware, 2003. Proceedings.*, pages 265–272. IEEE, 2003.
3. Daniel Mange, Andr´e Stauffer, and Gianluca Tempesti. Embryonics: A microscopic view of the molecular architecture. In *Evolvable Systems: From Biology to Hardware: Second International Conference, ICES 98 Lausanne, Switzerland, September 23–25, 1998 Proceedings 2*, pages 185–195. Springer, 1998.

4. Daryl Bradley, Cesar Ortega-Sanchez, and Andy Tyrrell. Embryonics+ immunotronics: A bio-inspired approach to fault tolerance. In *Proceedings. The Second NASA/DoD Workshop on Evolvable Hardware*, pages 215–223. IEEE, 2000.
5. Cesar Ortega-Sanchez, Daniel Mange, Steve Smith, and Andy Tyrrell. Embryonics: A bio-inspired cellular architecture with fault-tolerant properties. *Genetic Programming and Evolvable Machines*, 1:187–215, 2000.
6. Lucian Prodan, Mihai Udrescu, and Mircea Vladutiu. Self-repairing embryonic memory arrays. In *Proceedings. 2004 NASA/DoD Conference on Evolvable Hardware, 2004*, pages 130–137. IEEE, 2004.
7. Yuan Zhang, Youren Wang, Shanshan Yang, and Min Xie. Design of a cell in embryonic systems with improved efficiency and fault-tolerance. In *ICES*, pages 129–139. Springer, 2007.
8. Richard O Canham and Andy M Tyrrell. A hardware artificial immune system and embryonic array for fault tolerant systems. *Genetic Programming and Evolvable Machines*, 4:359–382, 2003.
9. Gayatri Malhotra, Joachim Becker, and Maurits Ortmanns. Novel field programmable embryonic cell for adder and multiplier. In *Proceedings of the 2013 9th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, pages 153–156. IEEE, 2013.
10. Gayatri Malhotra, AM Nagalakshmi, S Sudhakar, and Subramanya Udupa. An architecture of sequential embryonic cell for counter design. In *2014 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, pages 1–6. IEEE, 2014.
11. Zhai Zhang, QIU Yao, YUAN Xiaoliang, YAO Rui, CHEN Yan, and WANG Youren. A self-healing strategy with fault-cell reutilization of bio-inspired hardware. *Chinese Journal of Aeronautics*, 32(7):1673–1683, 2019.
12. Kasem Khalil, Omar Eldash, Ashok Kumar, and Magdy Bayoumi. Intelligent fault-prediction assisted self-healing for embryonic hardware. *IEEE Transactions on Biomedical Circuits and Systems*, 14(4):852–866, 2020.