A Novel SEU Self-Test Structure of SRAM-based Embryonic Electronic Cell

LI DANYANG, CAI JINYAN, ZHU SAI, MENG YAFENG
Department of electronic and optical engineering
Mechanical Engineering College
No.97, Heping West Road, Shijiazhuang
CHINA
ldy200607@126.com

Abstract: - SRAM-based embryonic electronic cell is susceptible to SEU (Single-Event Upset) in radiation space, which severely restricts its application on deep space. Based on the classic cell structure, delay comparison and dual-mode comparison are respectively used in configurable storage module and logic function module to design a novel embryonic electronic cell. The proposed method can effectively detect 1-bit and multi-bit SEUs in both configuration storage module and logic function module in real time. In addition, by using column elimination, the embryonic electronic cell array can achieve real-time self-repair in case of a fault and keep embryonic electronic array system working normally. Results of simulations in a 4-bit ripple carry adder verify the self-test ability of the embryonic electronic cell and self-repair ability of the embryonic electronic cell array.

Key-Words: - SRAM, embryonic electronic cell, self-test, delay comparison, dual-mode comparison, SEU

1 Introduction

Embryonics(Embryological electronics)[1][2] has drawn wide attention both at home and abroad because of its peculiar fault self-repair capability over the past few years. With the rapid development of electronic technology, the embryonic electronic array is confronted with the challenge of soft errors while coping with hard faults (stuck-at faults). The recent studies indicate soft errors mainly lead to faults of digital circuits. In the study on SRAM-based structure, the SEU problem has become a focus. Embryonic electronic array is a kind of SRAM-based array, which is similar to FPGA. The common embryonic electronic cell is easily subject to SEU because whose structures mainly contain SRAM-based memories and LUT. The occurrence probability of soft errors is far greater than that of hard faults, and such errors are difficult to be detected, located, and diagnosed. The gene memory module (configuration storage module) of the embryonic electronic array stores such configuration information as functions and connections of embryonic electronic cells. Errors if any may cause changes of cell functions and the array's topological structure, so as to faults of electronic cells. Due to the lack of effective means of self-test, the applications of embryonic electronic array system on deep space and strong electromagnetic radiation environment, etc. are severely restricted.

However, the design of a self-checking cell is a research difficulty. At present, most studies on the embryonic electronic system concentrate on cell structure[3-4], self-repair method[5-8], reliability analysis[9], etc. but few studies focus on self-checking structure. Because the structures of electronic cells in the embryonic electronic array should not be too complicated in order to reduce hardware cost and to reduce design effort. But simple structure cannot meet high fault coverage. Consequently, the self-checking structure of electronic cell is one challenge in the design of the embryonic electronic system.

In the existing designs of self-checking structures of electronic cells, Literature [10] introduces dual modular redundancy which is used to detect faults in MUX-based logic module and a special test sequence is added to detect faults in the configuration storage module before system configuration. In literature [11] two flip-flops and XOR Gate are used to detect stuck-at faults in the configuration memory module during configuration, and dual modular redundancy is used to detect faults in the logic module. Literature [12] introduces a method of adding antibody cells on the embryonic electronic array and comparison of data in the embryonic electronic cells with data in antibody cells to detect faults in the embryonic electronic cells. In Literature [13], hamming coding is used to detect soft errors in the configuration storage

module. In Literature [14], the extended hamming code is used to detect faults in the configuration storage module. In Literature [15], two kind of cell (including core cells and peripheral cells) constitute a prokaryotic electronic cell array by means of a bus structure and such array which is called “UNITRONICS ARCHITECTURE” realizes fault detection via peripheral cells.

In summary, the current self-checking structures used in embryonic electronic cells are still confronted with high hardware cost, complicated design and low fault coverage, etc. In this paper, based on the classic electronic cell structure, delay comparison and dual modular comparison are respectively adopted on the configuration storage module and logic function module to design a novel self-test structure which is capable of effectively detecting SEU. Such structure is not only simple for design, but also is able to detect 1-bit and multi-bit SEUs, with high fault coverage. The embryonic electronic array constituted by such electronic cells can achieve real-time test and self-repair, and ensure continually normal running.

2 The novel embryonic electronic cell structure

2.1 Main modules of the novel embryonic electronic cell

Modules of the novel embryonic electronic cell designed in this paper are shown in Fig.1. A kind of connection structure combining short connection with remote connection proposed in [3] is used on the embryonic electronic cell, mainly including Address Generator, Control Module, Switch Box, Self-Test Genes Module and Self-Test Function Module.

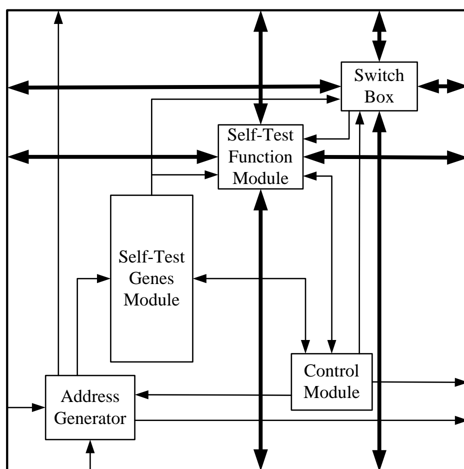

Fig.1 Modules of the Embryonic Electronic Cell

There have been a lot of detailed descriptions of modules such as Switch Box, Address Generator, so here, they are omitted for simplicity. The design of Self-Test Genes Module and Self-Test Function Module are mainly introduced as follows.

2.2 Design of self-checking configuration storage module based on delay comparison

The self-checking configuration storage module is shown in Fig.2. It mainly includes Gene Memory module, Delay module, and CMP module (the comparator). Address Generator in the electronic cell generates the cell address and configuration data are stored in Gene Memory module. The output gene configuration data is feed in CMP and simultaneously imported into the Delay module. Gene configuration data are delayed for one system clock, and sent to CMP. In CMP, the gene configuration data are compared with delayed gene configuration data. If the both are the same, the function is normal and self-test result is 0, otherwise, there is a fault and self-test result is 1. The comparison result is feed in Control Module.

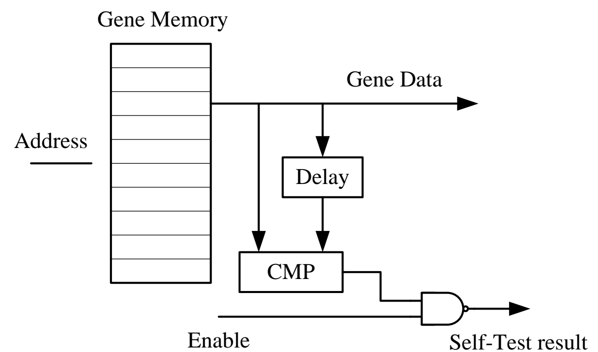

Fig.2 Self-checking Configurable Storage Module

In the self-checking Configuration Storage Module, the configurable storage data keep unchanged during normal running. So comparison of gene configuration data at two moments can effectively detect SEU occurred in the configurable storage data. Such self-checking structure is characterized with:

1) Universality and simple structure. Only some flip-flops and simple comparators are added. Delay module and CMP module are only related to gene length rather than gene memory scale, with low hardware cost. It can be used for self-test of any memory structures, featuring universality.

2) Low false alarm rate and high fault coverage. During self-test, SEU may also occur in Delay module. Since Delay module updates memory data in every system clock cycle, it has

extremely strong anti-SEU ability and the false alarm rate caused by SEU occurred in Delay module is very low. 1-bit and multi-bit SEUs occur in any position of Configurable Storage Module can be effectively checked, which has a very high fault coverage. Take the electronic cell designed in section 4 as an example, some comparisons of Hamming code and the method proposed are shown in Table 1. In Hamming code, there are more 6 parity bits in gene code, as memory cost is concert with the code length, the additional memory cost is about 10.53%. Also, there should be decoder and correcting circuits in every cell, which is complex, can be composed of a set of 2-input XORs and some ANDs and NOTs. Hamming code can correct single bit and detect double-bit SEU. In the proposed method, there is no additional memory cost, but more 57 Flip Flops. There is a comparator in every cell which can be composed of XORs and ORs. And any bit SEU can be detected. As is shown in Table 1, the proposed method is even better in terms of memory cost, logic gates cost and fault coverage, but needs more Filp Flops. More detailed results about Hamming code and time redundancy technique similar to this paper can refer to the literature [16][17].

Table 1 Comparison of the Two Methods

	Hamming code	Proposed method
Code length	63	57
Additional Memory cost	10.53%	0
Additional Flip Flops	0	57
Additional Logic gates	186 XORs and some ANDs and NOTs	57 XORs and 56 ORs
Fault coverage SEU bits≤2	100%	100%
Fault coverage SEU bits >2	0	100%

3) Self-test cannot work within the first clock cycle. Since CMP module still fails to receive the output from Delay module within the first clock cycle, self-test cannot be made. In the meantime, an Enable signal shall be added on CMP module and be controlled by Control Module in order to avoid the uncertain state produced within the first clock cycle.

2.3 Design of self-checking Logic Function Module based on dual-modular comparison

It is a classic design method to use dual-modular redundancy to achieve LUT self-test. Since the introduction of embryonic electronic array, such self-test structure with dual-mode comparison has been adopted by a lot of scholars. Such structure is mainly characterized with simple structure, high fault coverage, acceptable hardware cost, and real-time capability. The Logic Function Module designed in this paper is shown in Fig.3 and Fig.4. In the self-checking Logic Function Module, two 4-LUTs are used and their outputs are feed in CMP module, then CMP module compares the two outputs in real time. The probability of SEU occurrence in both LUTs at the same position, at the same time, is almost 0. Therefore, such self-test design with dual-modular comparison is capable of checking SEU occurred in LUT.

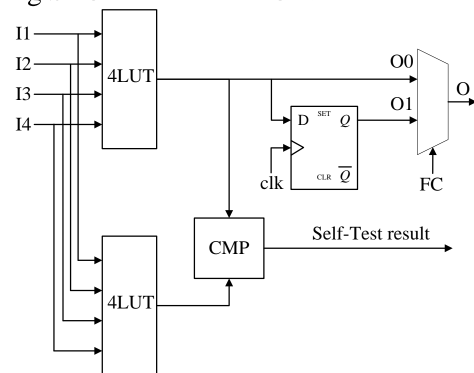

Fig.3 Self-checking Logic Function Module

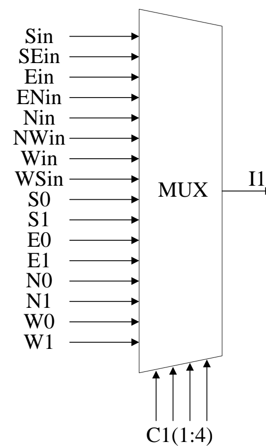

Fig.4 4-Input MUX

2.4 Control Module

Control Module designed in this paper is shown in Fig.5. CST is the self-test signal of the Configurable Storage Module and LST is the self-test signal of Logic Function Module. The two signals are connected to a D flip-flop via OR gate, as a clock input. When a fault is tested in the Configurable Storage Module or Logic Function Module, a clock signal of rising edge is generated

and D flip-flop is triggered to output “1”, representing a fault signal.

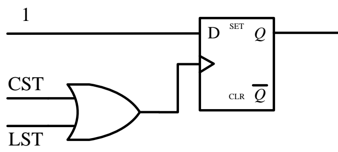

Fig.5 Control Module

3 Self-repair mechanism based on column elimination

Self-repair methods for embryonic electronic array mainly include single cell elimination and column (row) elimination. Column elimination is mainly used in this paper, as shown in Fig. 6.

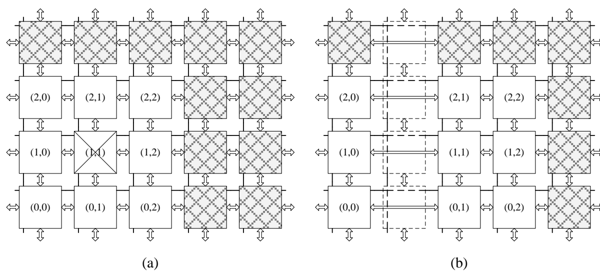

Fig.6 Column Elimination of Embryonic Electronic Cell Array

Single cell elimination can take full advantage of redundant embryonic electronic cell resources and enable the repair ability of the embryonic electronic array to achieve the theoretically maximum. However, every cell has to store the configuration data of the entire embryonic electronic array and the stored data presenting a square relation with array scale. Take the embryonic electronic array with n rows and n columns as an example, every cell has to store configuration data of n^2 cells. When the array scale increases to n+1, every cell has to store additional configuration data of 2n+1 cells, which consumes great storage resources. In addition, during single cell elimination, every electronic cell shall recalculate its address, consuming great calculation resources.

Compared with single cell elimination, main advantages of column (row) elimination include: every cell only should store configuration data of cells in the same row, substantially reducing resource consumption. Column (row) elimination is easily achieved with simple combinational circuit in every cell, which is rapid and does not require additional calculation resources. However, the idle electronic cells in the eliminated column (row) are also eliminated with column (row) elimination, causing resource waste and reducing the utilization rate of redundant cell resources. Therefore, in order to clearly explain resource utilization in column (row) elimination mechanism, SE (Self-repair Efficiency, SE) is defined as follows:

$$SE = \frac{fault_repaired}{idlecell} \times 100\% \qquad (1)$$

In Formula (1), *fault_repaired* represents the number of repaired faults and *idlecell* represents the number of idle cells. The theoretically maximum of SE is 1, meaning every idle cell in the array can repair one fault. Its minimum is 0, meaning the array cannot repair a fault. Comparison of single cell elimination with column (row) elimination is shown in Table 2.

Table 2 Comparison of Single Cell Elimination with Column (Row) Elimination

	Single cell elimination	Column (Row) elimination
Number of data stored	n^2	n
Elimination realization	additional calculation resources	Simple combinational circuits
Self-repair efficiency	Theoretical maximum	Low utilization rate of redundant cell resource

4 Simulation and experiments

4.1 Cell model

The simulation experiment runs on XILINX ISE Design Suite 12.2 and ISM. During simulation, fault injection is adopted to verify the self-test ability and self-repair ability of the electronic cell array. The package structure of the electronic cell designed in the experiment is shown in Fig. 7.

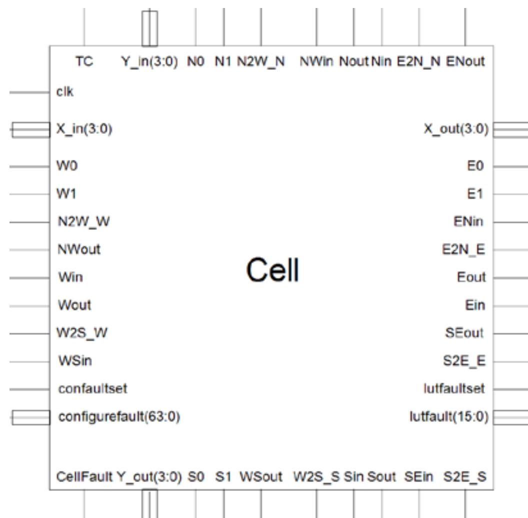

Fig.7 Package Structure of the Electronic Cell

In the Fig7, clk is a clock input port. X_in(3:0), Y_in(3:0), X_out(3:0), and Y_out(3:0) are address input and output ports. CellFault is the output port of cell self-test result, through which the cell transfers the self-test result to the below cell in the same column. TC is the state receiving port, which receives the self-test result from the above cell in the same column so as to control cell state. For example, when the self-test result of the above cell is a fault, the faulty cell becomes "transparent" and CellFault outputs a self-test fault signal and cooperates with TC to achieve column elimination. confaultset is fault injection set-port of the Configurable Storage Module. When confaultset is 1, a fault starts to be injected. When confaultset is 0, no fault is injected. configurefault is a fault injection port of the configuration data. lutfaultset is fault injection set-port of the Logic Function Module. When lutfaultset is 1, a fault starts to be injected. When lutfaultset is 0, no fault is injected. lutfault is a fault injection port of the Logic Function Module. Electronic cells are connected both in a short distance and a remote distance. Eight signals from S0 to W1 are remote connection signals input via switch box and the eight signals from Sin to WSin are short connection signals input surrounding the electronic cell via a straight-through cable.

Configuration data of every electronic cell mainly includes LUT logic, LUT delay set, LUT input selection, and remote connection, which is as shown in Table 3.

Table 3 Electronic Cell Configuration Data

Configuration	56-33	32-17	16	15-0
Function	Remote connection	LUT input	Delay	LUT logic

Simulation realization is made with a 4-bit ripple carry adder. 4-bit ripple carry adder is constituted by four 1-bit full-adders through cascading, with the schematic circuit diagram as shown in Fig.8.

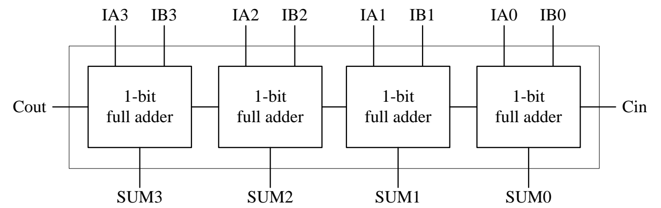

Fig.8 4-bit Ripple Carry Adder

In the simulation, a 5-row 4-column electronic cell array is used to realize function of 4-bit ripple carry adder. The relationship between input and output of all cells are shown in Fig. 9.

In Figure 9, A3~A0 and B3~B0 are two 4-bit inputs, CIN is carry input, SUM3~SUM0 are 4-bit outputs, and COUT is carry output. All input signals are feed in the electronic cell array via remote connection (S0~W1) and short connection (Sin~WSin). The logic output of every electronic cell is sent to other electronic cells in the same way and all electronic cells cooperate with each other to realize system functions.

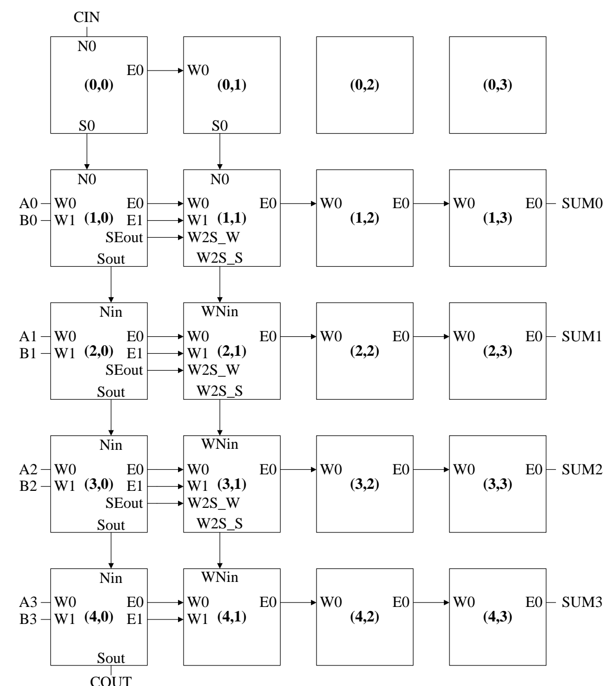

Fig.9 Schematic Diagram of Main Connection Relations of the Electronic Cell Array

Configuration data of every electronic cell is shown in Table 4.

Table 4 Configuration Data of the Electronic Cell

(0,0) 01FFFFF40 0000000	(0,1) 01FFFFF80 0000000	(0,2) 000000000 0000000	(0,3) 000000000 0000000
(1,0) 01FFED7FD DF8E8E8	(1,1) 01FFFD7F DDF89696	(1,2) 01FFFD7E 00000000	(1,3) 01FFFD7E 00000000
(2,0) 01FFED7FD DE8E8E8	(2,1) 01FFFD7F DDE89696	(2,2) 01FFFD7E 00000000	(2,3) 01FFFD7E 00000000
(3,0) 01FFED7FD DE8E8E8	(3,1) 01FFFD7F DDE89696	(3,2) 01FFFD7E 00000000	(3,3) 01FFFD7E 00000000
(4,0) 01FFED7FD DE8E8E8	(4,1) 01FFFD7F DDE89696	(4,2) 01FFFD7E 00000000	(4,3) 01FFFD7E 00000000

4.2 Fault injection experiments

4.2.1 Fault injection experiment of a single cell

Take an electronic cell with the coordinate of (1,0) as an example, LUT logic function unit and configurable storage unit are respectively injected with a fault, and simulation results are respectively shown in Fig.10 and Fig. 11.

In Figure 10, X_in and Y_in are address input signals of the electronic cell, and X_out and Y_out are address output signals. Ain, Bin are two low order input signals and Cin is carry input signal of 4-bit ripple carry adder. A0 and B0 are two input signals (Ain and Bin) from W (West) direction, N0 is the input signal (Cin) from N (North) direction, E0 and E1 are two remote output signals toward E (East) direction. Sout is the output of the adjacent cell of the electronic cell toward S (South) direction and SEout is the output of the electronic cell toward an adjacent cell in SE (SouthEast) direction. TC is the state input signal of the adjacent cell. CellFault is the fault self-test result signal of the electronic cell. clk is the system clock signal. lutfaultset is LUT fault set signal, and lutfaut is the LUT fault to be set.

During 0-100ns, the cell function is to calculate whether the adding of three (0,1,1) input from Ain, Bin, and Cin produces a carry signal, the logic output "1" is feed in the S direction cell and SE direction cell via Sout and SEout ports, and two input signals are sent to the W direction cell from E0 and E1 ports to W0 and W1 ports. At 100ns, when a fault is injected into LUT logic unit, its fourth bit changes from 1 to 0, lutfaultset signal is activated, and CellFault signal becomes a high level, representing the cell detects a fault. X coordinate and Y coordinate of the cell automatically reduce by 1 and the address becomes (0,-1). At this moment, the electronic cell is "transparent" and the logic output is under an uncertain state. This indicates the electronic cell is capable of testing a SEU fault occurred in LUT.

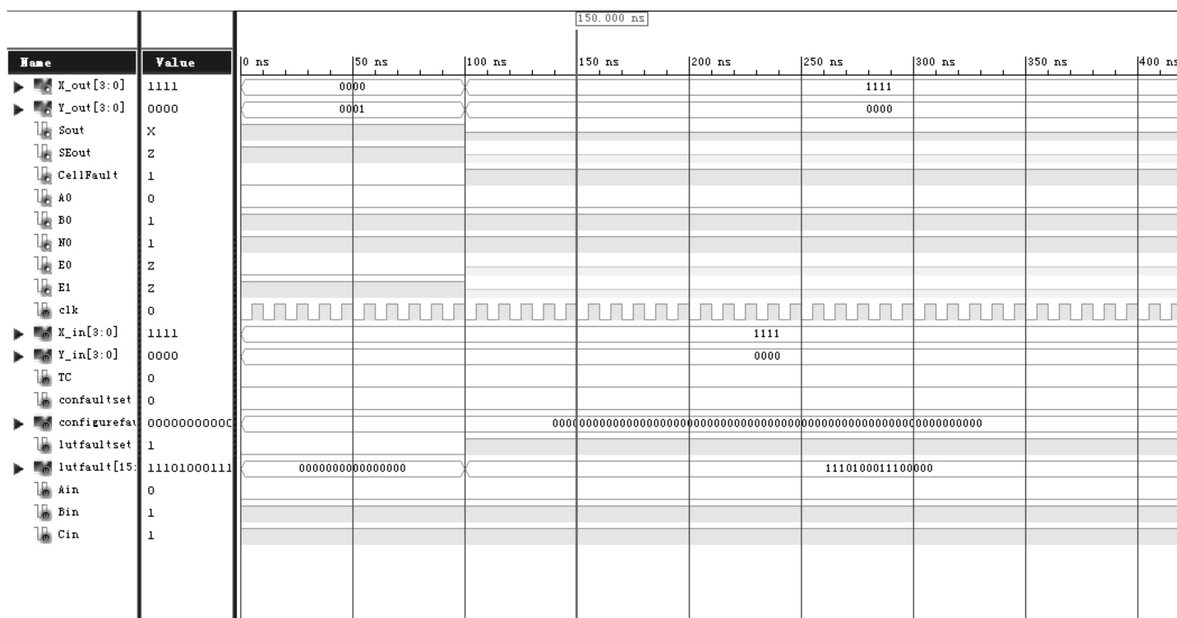

Fig.10 LUT Fault Injection Experiment

In Figure 11, configureset is the fault setting signal of the Configurable Storage Module and configurefault is the fault data injected into the Configurable Storage Module. At 50ns, configureset is activated, the fault injected into the Configurable Storage Module is 64'H01FFFD7FDDF8E8E8, which means the state of the bi-directional port E1 changes from output port to input port, and CellFault

signal becomes a high level, representing the electronic cell detects a fault, the cell address becomes (0，-1) and E1 becomes a high resistance “Z”. This proves the electronic cell is capable of checking a SEU fault occurred in the Configurable Storage Module

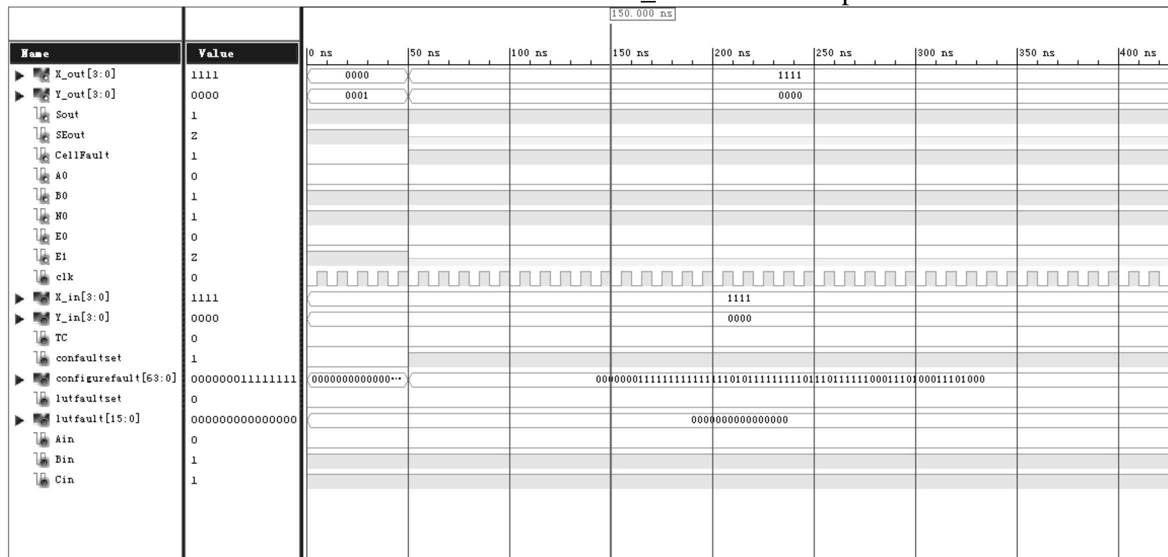

Fig.11 Fault Injection Experiment of Configurable Storage Module

4.2.2 Fault Injection Experiment of the Electronic Cell Array

In Fig.12, a3~a0, b3~b0 are two 4-bit input signals, and cin is a 1-bit carry input. sum3~sum0 are 4-bit output signals and cout is 1-bit carry output. X_in and Y_in are the input address of the electronic cell with the coordinate of (0,0), and X_out and Y_out are the output address of the electronic cell with the coordinate of (4,3) in the last row and the last column. During 0-100ns, the array operates randomly selected carry addition operations with the inputs of (A3~A0)1010, (B3~B0)0101, and (CIN)1. During 100-200ns, the array operates carry addition with the inputs of (A3~A0)1001, (B3~B0)1100, and (CIN)1. During 200-300ns, the array operates carry addition operations with the inputs of (A3~A0)0101, (B3~B0)0111, and (CIN)0. At 50ns, lutfaultset changes into a high level, a fault is injected into LUT module, the value of X_out changes from 3 to 2, indicating column elimination occurs in the electronic cell array but the electronic cell array keeps working normally. At 150ns, confaultset signal becomes a high level, a fault is injected into the Configurable Storage Module, the value of X_out changes from 2 to 1, indicating column elimination occurs in the electronic cell array but the electronic cell array keeps working normally. This indicates the electronic cell array can detect SEU faults occurred in the LUT and Configurable Storage Module in real time, and realize real-time self-repair, so as to maintain the normal running of the electronic cell array.

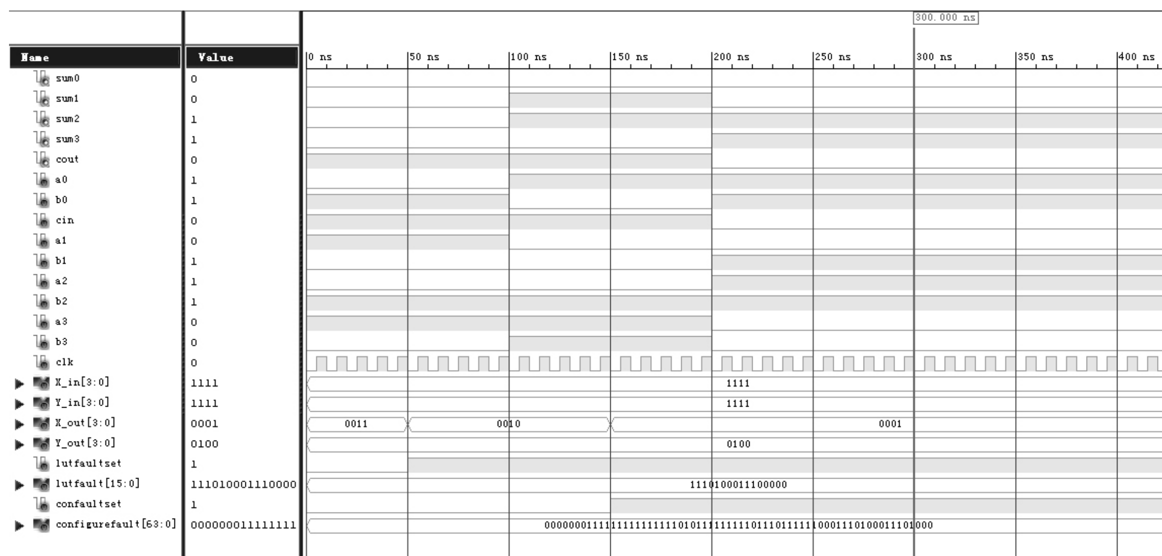

Fig.12 Fault Injection Experiment of the Electronic Cell Array

5 Conclusions

In this paper, an effective self-test method on the SRAM-based embryonic electronic cell is studied in case of SEU and a novel embryonic electronic cell structure is proposed. The electronic cell model is built under XILINX ISE 12.2 to design Address Generator, Switch Box, Configurable Storage Module, Logic Function Module, and Control Module, etc. In addition, delay comparison and dual-modular comparison, etc. are respectively used on Configurable Storage Module and Logic Function Module. The 4-bit ripple carry adder is used as an example and fault injection is used to simulate SEU impact on the electronic cell. Simulation results verify the proposed method can effectively detect SEU and the self-repair mechanism with column elimination is able to achieve real-time self-repair.

As a whole, the proposed method is characterized with simple structure, low hardware cost, and high fault coverage. Consequently, the method proposed in this paper has some application prospects, providing a new thought for the application of the embryonic electronic array on the radiation environment.

References:

[1] Hugo de Garis. Genetic Programming: Artificial Nervous Systems, Artificial Embryos and Embryological Electronics [C]. Proceedings of the 1st Workshop on Parallel Problem Solving from Nature. 1990:117-123.

[2] Daniel MANGE, Serge DURAND, Eduardo SANCHEZ, et al. A New Paradigm for Developing Digital Systems Based on a Multi-Cellular Organization[C].IEEE International Symposium on Circuits and Systems. 1995:2193-2196.

[3] Sai Zhu, Jinyan Cai, Yafeng Meng, Danyang Li, Junbin Zhang, Tianzhen Meng. A Novel Structure of Embryonics Electronic Cell Array[J]. WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS. 2014, 13:224-232.

[4] M. Samie, E. Farjah, G. Dragffy. Cyclic metamorphic memory for cellular bio-inspired electronic systems[J]. Genet Program Evolvable Mach, 2008, 9:183-201.

[5] S. M. Achasova. Self-Replicating Structure as an Artificial Multicellular Organism[J]. Cybernetics and Systems Analysis, 2014, 50:316-323.

[6] Can Ozturkeri, Colin G. Johnson. Self-repair ability of evolved self-assembling systems in cellular automata[J]. Genetic Programming and Evolvable Machines, 2014, 15:313-341.

[7] Isaak Yang, Sung Hoon Jung, Kwang-Hyun Cho. Self-Repairing Digital System With Unified Recovery Process Inspired by Endocrine Cellular Communication[J]. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEM, 21:1027-1040.

[8] M.Aarthi, S. Karthigailakshmi. Modeling of Bio-Inspired Self-Repairing Digital System[J]. IOSR Journal of Electronics and Communication Engineering, 2014, 9:29-36.

[9] Zhai Zhang, Youren Wang. Method to self-repairing reconfiguration strategy selection of embryonic cellular array on reliability analysis[C]. 2014 NASA/ESA Conference on Adaptive Hardware and Systems. 2014: 225-232.

[10] Daniel Mange, Andre Stauffer, and Gianluca Tempesti. Embryonics: A Microscopic View of the Molecular Architecture[C]. Proceedings of the Second International Conference on Evolvable Systems: From Biology to Hardware. Lecture Notes in Computer Science. 1998:185-195.
[11] SESAR ORTEGA-SANCHEZ, DANIEL MANGE, STEVE SMITH, et al. Embryonics: A Bio-Inspired Cellular Architecture with Fault-Tolerant Properties[J]. Genetic Programming and Evolvable Machines, 2000, 1:187-215.
[12] Daryl Bradley, Cesar Ortega-Sanchez, Andy Tyrrell. Embryonics + Immunotronics: A Bio-Inspired Approach to Fault Tolerance[C]. Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware. 2000:215-223.
[13] Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. Self-Repairing Embryonic Memory Arrays[C]. Proceedings of the 2004 NASA/DoD Conference on Evolution Hardware. 2004:130-137.
[14] Yuan Zhang, Youren Wang, Shanshan Yang, Min Xie. Design of a Cell in Embryonic Systems with Improved Efficiency and Fault-Tolerance[C]. Proceedings of the Second International Conference on Evolvable Systems: From Biology to Hardware. Lecture Notes in Computer Science. 2007:129-139.
[15] Mohammad Samie, Gabriel Dragffy, Tony Pipe. Bio-Inspired Self-Test for Evolvable Fault Tolerant Hardware Systems[C]. 2010 NASA/ESA Conference on Adaptive Hardware and Systems. 2010:325-332.
[16] Hentschke R. , F. Marques, F. Lima, et al. Analyzing Area and Performance Penalty of Protecting Different Digital Modules with Hamming Code and Triple Modular Redundancy[C]. Proceedings of 15th Symposium on Integrated Circuits and Systems Design, 2002:95-100.
[17] Lorena Anghel, Michael Nicolaidis. Cost reduction and evaluation of temporary faults detecting technique[C]. Proceedings of the conference on Design, automation and test in Europe, 2000:591-598.