A Realisation of Fault Tolerant ALU-With TMR Method

Jaimini patel, Prof. Deepali H. shah
L.D. College of Engineering,
Ahmedabad

***Abstract* - This paper represents the fault tolerant ALU with Triple modular redundancy on FPGA. TMR technique is mitigating the single error upsets of the module. TMR method gives fault tolerant result but with penalty of area. TMR technique is used in aviation application and space application where radiation effects are takes place.**

***Keywords* - Fault Tolerant, ALU, TMR(triple modular redundancy), FPGA, Xilinx ISE**

I. INTRODUCTION

In recent years, working on reliable communication transmission plays a dynamic role. So that a single error may shutdown the whole system and give rise to erroneous data[17]. System reliability is one of major issues in embedded processors schemes for space application such as satellite, military, communications and etc. Various attacks exist in space on integrated circuits that comes from sun activity[17]. Such as solar rays which are composed of charged particles. The radiation from sun effects in integrated circuits make digital damage and upsets such as SEU (Single Event Upset), SET (Single Event Transient) and etc as presented in [1],[7]. Such attacks can upset either combinational logic or sequential logic. In other words a bit flip can occur in register bits and if one bit of main storage system is changed the mission of system would be completely different. In such scenario the error control or fault tolerant methods are employed to keep integrated circuits against these attacks in space. To achieve such purpose we consider error detection and correction codes (EDAC) method. We have designed a 8-bit ALU. Our 8-bit ALU model consists of the following function units: Arithmetic operation consist of Full-Adder and subtractor. Bitwise logic operation such as: XOR, AND, OR, and NOT. Bit-shifting operations such as: shifting to the left or right, comparisons.

Motivation

Our goal is to designing a new 8-bit ALU that is secure beside many attacks or faults and able to correct bit fault in any position of its 8-bits input register of ALU [2]. Because the radiation effects on electronic circuits may cause to be inverted data bits of registers or memories[15]. If one bit of main storage system is changed the mission of system would be completely different. The high motivation in choice of TMR is that, it is able to correct single errors upsets and gives the correct output.

II. INTRODUCTION OF FAULT TOLERANT TECHNIQUE - TMR

Many methods have been proposed to mitigate the effects of single event upset (SEU) in FPGAs. TMR (Triple Modular Redundancy) is one of popular technique to mitigate SEU in FPGA[14]. TMR involves creating three redundant copies of a circuit and adding majority voters to select the correct circuit output from the three copies. With this mitigation methodology, a single module failure will not cause an error in the circuit output, since the other two modules continue to operate correctly and will overrule the faulty module. TMR is thus often joined with configuration scrubbing in attempt to prevent multiple upsets from affecting the circuit at one time.[13]

Static upsets in the configuration memory are not necessarily synonymous with a functional error; however, soft-errors are by definition a functional error. Upsets might or might not have an effect on functionality. However, an accumulation of upsets in the configuration memory is eventually certain to lead to a functional failure. Design mitigation techniques, such as triple module redundancy, can harden functionality against SEUs and SETs, while the SEUs are corrected so that static-errors do not accumulate and soft-errors do not propagate. [14] Implementing triple redundant circuits in other technologies, such as ASICs, is traditionally limited to protecting only the flip flops of the user's design from SEU, because logic paths in between the flop-flops are typically hard-wired, non-reconfigurable gates. For such fixed logic technologies, this is adequate protection from SEUs, but can still leave the circuitry vulnerable to SETs. For a technology that is vulnerable to SETs, further protection can be achieved through full module redundancy. Full module redundancy is the required implementation of TMR in FPGAs, because all the logic paths, not just the flip flops, are susceptible to SEUs. This means that three full copies of the base design will be implemented to protect circuit functionality from SEUs, as well as SETs. [14]

The insertion of voters is an important aspect of applying TMR to a design. One of the more challenging design problems is determining where to insert synchronization voters. Synchronization voters are used to keep the sequential logic state of the three redundant copies of a circuit (domains) synchronized when there are SEUs that affect design feedback[15]. In order to keep a design synchronized properly, synchronization voters must be inserted in enough locations to intersect. TMR is often implemented by hand, and the process of properly Inserting synchronization voters manually can be tedious and error-prone.[15]

The basic concept of triple redundancy is that a sensitive circuit can be hardened to SEUs by implementing three copies of the same circuit and performing a bit-wise "majority vote" on the output of the triplicate circuit. See Figure 1.

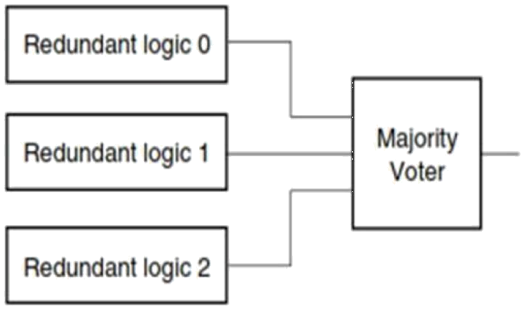

Figure:1 Triple Redundancy with Majority Voter [13]

The circuit in question can be a mere flip flop or an entire logic design. The function of the majority voter is to output the logic value ("1" or "0") that corresponds to at least two of its inputs. For example, if two or more of the voter's three inputs are a "1," then the output of the voter is a "1." If the inputs of the voter are labelled A, B, and C, and the output V, respectively, then the Boolean equation for the voter is: V= AB + AC + BC. The Truth-Table is shown in Table 1. [13]

The logic gate representation of the majority voter is shown in Figure 2

Table 1: Majority Vote Truth-Table [13]

A	B	C	V
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

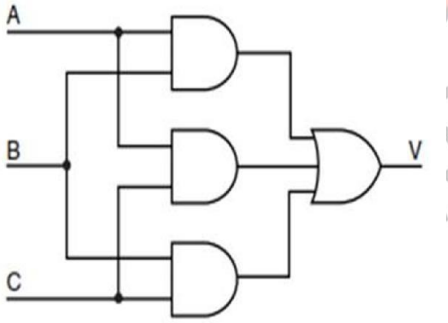

Figure 2: Majority Voter[13]

ALU with fault tolerant technique triple modular redundancy

The other work in such field is designing a 32-bit ALU using Triple Modular Redundancy (TMR) with single voting. In fact, it uses hardware redundancy technique in the combinational logic and allows voting the correct output value in the presence of faults[7]. The majority voter scheme is depicted in Fig. 3.

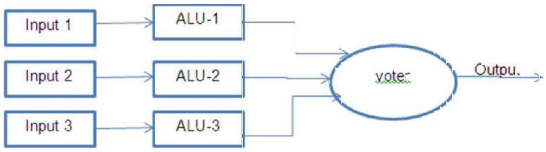

Figure 3.Triple modular redundancy [15]

The concept of TMR is devised firstly by Von Neumann. The circle Voter in Fig. 4 is called a majority organ by Von Neuman. In TMR technique the logic is triplicated in the output, the voters identify the correct value. The entire scheme is shown in Fig. 3. In this technique all registers should be tripled to protect circuits against radiation effects. The voter must be added in the output [15]. The error will not be reflected in the output of voter if, one component fails. The hardware overhead in TMR is the addition of the two registers of the same size. Moreover there exist n voters for each n-bit register. So, such method is also need extra two spare 8-bit ALU and it is give rise to 200% hardware. Triple modular redundancy scheme is more effective according to area and performance to preserve registers and small memory structure[14].

III. SIMULATION RESULTS

Here Redundancy module means triplicated module of ALU is performed here. After applying triple modular redundancy technique on ALU it gives better result than without TMR.TMR ALU is operated as like as simple ALU. ALU module is triplicated. We apply faulty input from outside. Then results are compared with each other. We will get correct result from that triplicated result. TMR ALU take majority voter output as a correct output. Here three outputs (result, result_1, result_2) of three modules are generated. Than they are compared:

q1=0; k1=1; than result=result_1; so here result_2 is faulty q2=0; K2=1; than result_1=result_2; so here result is faulty q3=0; k3=1; than result=result_2; so here result_1 is faulty Output will reconfigured
and display as output a1.

Example: operation ADDITION

A = 01010101 ; ALU 1B
=00000011

Result=01011000

A_1=10101010 ; ALU 2 // faulty
input is given here

B_1=00000011

Result_1=10101101

A_2=01010101 ; ALU 3

B_2=00000011

Result_2=01011000

Q3=00000000;

K3=1;

So result and result_2 are equal. Means ALU2 has fault.

After operation A1=10101101.

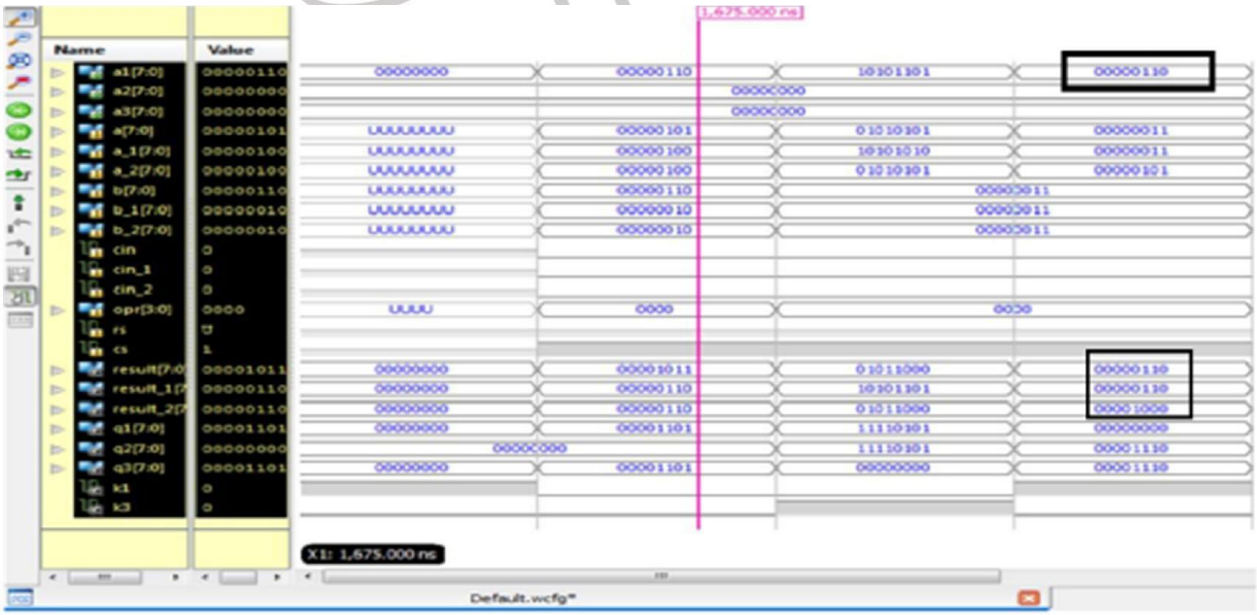

Figure:4 Simulation result of Addition operation

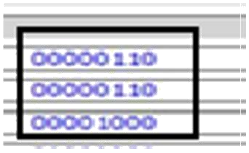

Figure 5. tmr output with faulty output

Figure 6. final output

IV. CONCLUSION

ALU module should be triplicated for triple modular redundancy technique. Using VHDL coding on Xilinx they are realized on ISE tool software. Triple modular redundancy corrected faulty output. And gives corrected result as an output. Majority voter scheme select the correct output and faulty one is discarded. We can see it in addition simulation result. We also can check this for other application of ALU: XOR, AND, subtraction, comparison operation etc..

REFERENCES

[1] MihalisPsarakis, Andreas Apostolakis, "Fault Tolerant FPGA Processor Based on Runtime Reconfigurable Modules" 17th IEEE European Test Symposium (ETS),2012
[2] Prof.S.Kaliamurth, Ms.U.Sowmmiya "VHDL Design of FPGA Arithmetic Processor" 2011 Global Journals Inc. (US), pg no 31-35
[3] Chuan Hong, Khaled Benkrid, XabierIturbe, Ali Ebrahim "Design An Implementation Of Fault-tolerant Soft Processors On Fpgas" IEEE 2012
[4] Martin Straka, Jan Kastil, ZdenekKotasek "Modern Fault Tolerant Architectures Based on Partial Dynamic Reconfiguration in FPGAs" IEEE 2010, 173-176
[5] JASON A. CHEATHAM and JOHN M. EMMERT "A Survey of Fault Tolerant Methodologies for FPGAs" ACM Transactions on Design Automation of Electronic Systems, Vol. 11, No. 2, April 2006, Pages 501–533.
[6] Edward Stott, Pete Sedcole, Peter Y. K. Cheung "FAULT TOLERANT METHODS FOR RELIABILITY IN FPGAs" IEEE 2008, pg no. 415-420
[7] Khaled Elshafey "FAULT-TOLERANT FPGA-BASED SYSTEMS" Computing and Informatics, Vol. 21, 2002, 489–505
[8] Pieter Anemaet , Thijs van As "Microprocessor Soft-Cores: An Evaluation of Design Methods and Concepts on FPGAs"
[9] Bryan H. Fletcher "FPGA Embedded processors: reveling True system performance" embedded systems conference, sanfrancisco, San Francisco, 2005
[10] Ms. Shweta s. mesharm , Ms. UJwala A. Belorkar " design approach for fault tolerance in FPGA architecture" international journal of VLSI design & communication systems , vol.2 no.1, march 2011, pg no.87-95
[11] Mrsjamuna.S, Dr. V.K.Agrawal, "Fault tolerant techniques for reconfigurable devices: a brief survey" IJAIEM, vol-2, january 2013 pg no 339334
[12] Prof. R.V.Kshrisagar, Sanjeev Sharma "An Algorithm for fault tolerance in FPGA" European scientific journal, august 2013, vol-9, pg no. 334342
[13] Carl Carmichael "Triple Module Redundancy Design Techniques for Virtex FPGAs" XAPP197 (v1.0.1) July 6, 2006, xilinx .com
[14] Jonathan Johnson Michael Wirthlin "Voter Insertion Techniques for Fault Tolerant FPGA Design" NSF Center for High Performance Reconfigurable Computing (CHREC) Dept. of Elec. & Comp. Engineering Brigham Young University Provo, UT 84604
[15] Brian Pratt, Michael Caffrey, Derrick Gibelyou, Paul GrahaBrian Pratt, Michael Caffrey, Derrick Gibelyou, Paul Graham "TMR with More Frequent Voting for Improved FPGA Reliability"
[16] Ian Kuon, Russell Tessier, and Jonathan Rose "FPGA Architecture: Survey and Challenges" Foundations and Trends In Electronic Design Automation Vol. 2, No. 2 (2007) 135–253
[17] Jaimini S. Patel, Deepali H. Shah "Different Types of Fault Tolerant Techniques of Softcore Processor" I-journal , issn no 2347- 4890 volume 2 issue 3, march 2013.