Volume 5, Issue 2, February 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Light Weight Novel Approach Voter Design Using Sic Vector in TMR Systems

V. Abuhanifa, D. John Pragasam, Dr. R. Ganesan
Sethu Institute of Technology
Tamil Nadu, India

Abstract--The main objective of this project is to the majority voted redundancy is increasingly implemented in fault-tolerant design simulator using fault injection experiments. The new Enhanced Lockstep scheme requires significantly shorter error recovery time compared to conventional lockstep scheme and uses significantly smaller number of slices compared to known TMR based. Finally the novel test pattern generator (TPG) for built-in self-test is obtained and the efficiency of the proposed approach was validated through fault injection experiments.

Keywords— Test pattern generator (TPG), triple modular redundancy (TMR), Fault-tolerant design, Xilinx ISE 12.1i software.

I. INTRODUCTION

Fault redundancy is one of the fundamental aspects of the methodologies to perform partitioning TMR insertion to reduce SEUs in the FPGA logic paths are presented in this paper. It is proved that the maximal probability of two simultaneous errors decreases dramatically with the number of logic partitions in the TMR designs [1] These techniques should allow for online error detection or/and correction during system operation, very fast fault location, quick Recovery from temporary failures, and fast permanent fault repair through reconfiguration[2].utilizing different redundancy configuration and voter insertion algorithm to observe variation in these performance factors for FPGA design [3].

II. LITERATURE SURVEY ON VOTER DESIGN TMR SYSTEMS

The TMR techniques are interested in designing and implementing a fault-tolerant (FT) soft core processor using Virtex-5 FPGA**.**

1) Reliability: Reliability is a major concern for electronic circuits, especially for those that operate in harsh environments. One source of problems are Single Event Upsets (SEU), which change the value of flip flops and memory cells. SEUs are a major issue for SRAM based Field Programmable Gate Arrays (FPGAs), as they may alter the circuit functionality, creating errors that will only be removed if the device is reprogrammed. The cost of traditional techniques to deal with SEUs, like triplication, can be excessive in some applications [1].

2) Single-Event Upset: The major effects caused by them are known as Single-Event Upsets (SEUs) or soft errors, because only some logic state(s) of memory element(s) are changed but the circuit/device it is not permanently damaged is called Single-Event Upsets [3].

3) Triple-modular redundancy: The Xilinx Vertex family of static random access memory (SRAM) based field programmable gate array (FPGA) devices have made inroads into space-based computational platforms over the past decade. These devices are well-suited for digital signal processing (DSP) algorithms that are often used on orbit, providing the speedup of custom hardware without the cost of fabricating an application-specific integrated circuit (ASIC). Triple modular redundancy (TMR) is a widely used mitigation technique to protect FPGA circuits against single event upsets (SEUs) [4]. TMR, however, does not adequately protect signals that cross asynchronous clock domains. Signals which cross clock domains in TMR circuits may suffer from the combined effects of two failure modes: asynchronous sampling effects and SEUs [2].

4) Fault Injection: In order to evaluate the detection and recovery coverage of the ScTMR processor we have carried out statistical fault injection experiments. The experiments have been performed using the FPGA-based fault injection technique [7].

5) Enhanced Lockstep Scheme: The architecture of the fault-tolerant system proposed here, whose two main blocks are: the Enhanced Lockstep scheme and the fault-tolerant Configuration Engine [10]. It relies on using two Xilinx Vertex hardware primitives The COMP_MUX consists of two blocks:

1) The Comparator that indicates any mismatch between the outputs Out1 and Out2 of DUT1 and DUT2.

2) The Multiplexer which connects one of the processors to the system output, so that if one of them is reported to be faulty, the other is switched on.

The switching is an atomic operation executed in one clock cycle. Once the error is localized by the FT Configuration Engine. The affected processor is reconfigured to eliminate its configuration upset. Then, the two processors need to be synchronized to put the newly reconfigured one to the same state as the correct one, thus enabling them to continue executing the same task in Lockstep again [5]. The recovery process of the Enhanced Lockstep scheme is handled by the Context Recovery Block (CRB).

III. PROPOSED METHODS

To validate the efficiency of the fault-tolerant approach proposed, we have provided the FT Configuration Engine with the possibility to carry out automatic fault injection campaigns for the configuration memory of P1 and the COMP MUX.

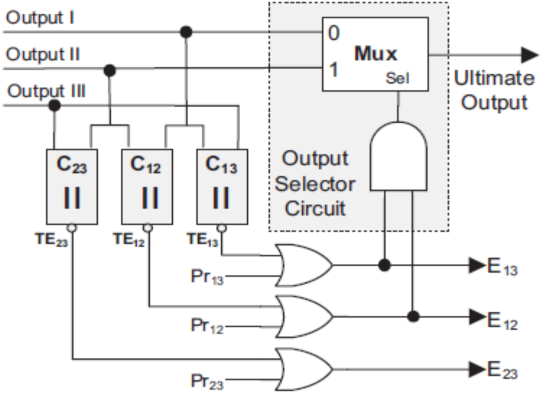


Fig.1. Proposed voter

The goal is to obtain some statistical data on the design robustness, measured by the percentages of sensitive bits and persistent errors once a fault is injected, the lock stepped pair of processors P1 and P2 executes the application program that checks whether a peripheral of a soft core operates correctly. This application was chosen; because it involves a relatively large number of transactions passing through PLB bus and thus offers the advantage that COMP MUX can verify the consistency of outputs of P1 and P2 for frequently varying data. The architecture of the proposed voter is depicted in Fig. 3 As shown in the figure; three comparators (*C*12, *C*13, and *C*23) are used to represent any mismatch between TMR modules. As an example, *T E*23 signal is activated once a mismatch between Outputs II and III is detected. If one of the modules generates an erroneous output (e.g., Output I), two of the comparators (here, *C*12 and *C*13) will activate the mismatch signals (here, *T E*12 and *T E*13) and only one of the comparators (here, *C*23) will not activate the corresponding mismatch signal (here, *T E*23). In case of a faulty comparator (e.g., *C*13), only the corresponding signal (here, *T E*13) is activated and the other signals (here, *T E*12 and *T E*23) are deactivated.

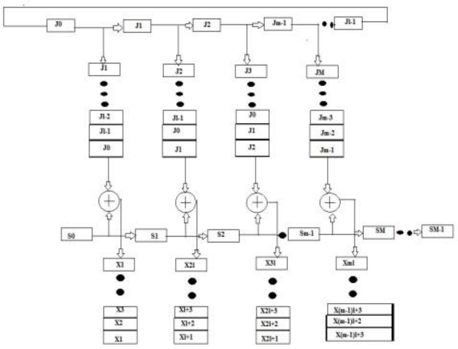


Fig.2. SIC vector generation

This paper has proposed a low-power test pattern generation method that could be easily implemented by hardware. It also developed a theory to express a sequence generated by linear sequential architectures, and extracted a class of SIC sequences named MSIC as fig 2 shows. Analysis results showed that an MSIC sequence had the favorable features of uniform distribution, low input transition density, and low dependency relationship between the test length and the TPG's initial states. Combined with the proposed reconfigurable Johnson counter or scalable SIC counter, the MSIC-TPG can be easily implemented, and is flexible to test-per-clock schemes and test-per-scan schemes. To analyze the fault injection results, the effect of each fault injection is classified as follows.

a) Overwritten: A fault is overwritten before it is propagated to the module outputs. Thus, the fault has no effect either on the output of the running workload or on the processor states at the end of the workload execution.

b) Latent: A fault does not affect the workload output but it does cause a mismatch in the processor state at the end of the workload execution. **c) Corrected:** An error is detected, located, and corrected by the employed technique.

IV. RESULTS

The performance of the scheme with existing scheme is analyzed based on the area and time consumption which was given in table 1.Based on the analysis, it is come to known that the testing process which uses SIC technique gives the better reduction in area and time when compared to the other methods The table consist of four columns as they depend the input output block, time and the look up table slices, input LUT. The parameter is displayed in the table I as refer.

The proposed technique is determined the look up table slice, input and the input output block of that number. So this is also achieved.

Table I **L**ist **o**f **p**arameter **w**ith **t**he **c**omparison **o**f **e**xisting **m**ethod **a**nd **p**roposed **m**ethod

Parameter	Existing Method	Proposed Method
Number of Slice LUTs	17	16
Number of 4 input LUTs	31	28
Number of bonded IOBs	50	46
Time (ns)	15.9	14.9

The area and time comparison of existing and proposed technique as shown below,

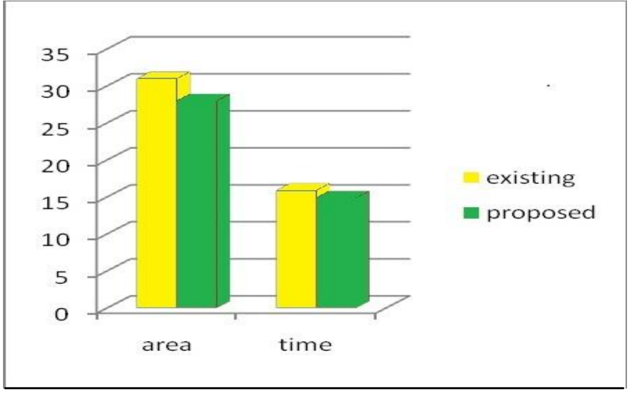
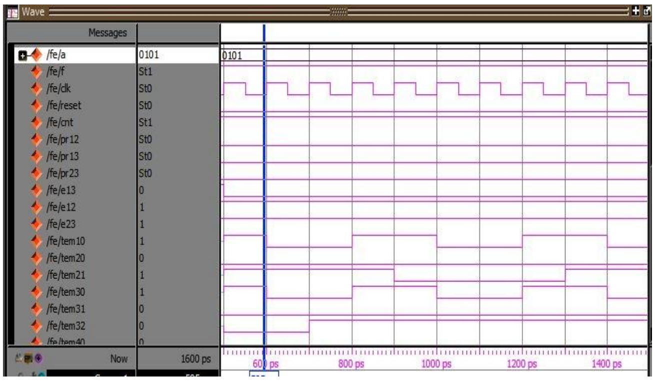


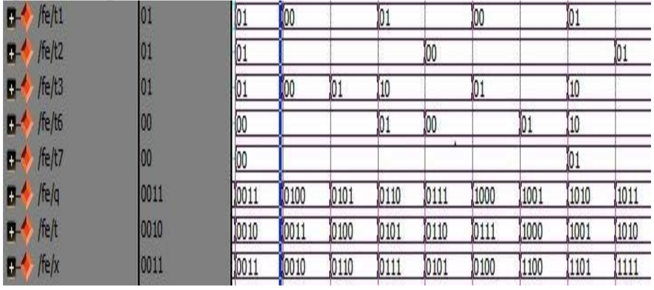
Fig.3. Area and Time Comparison



```
/fe/a       0101
/fe/f       St1
/fe/clk     St0
/fe/reset   St0
/fe/cnt     St1
/fe/pr12    St0
/fe/pr13    St0
/fe/pr23    St0
/fe/e13     0
/fe/e12     1
/fe/e23     1
/fe/tem10   1
/fe/tem20   0
/fe/tem21   1
/fe/tem30   1
/fe/tem31   0
/fe/tem32   0
/fe/tem40   0
Now         1600 ps
```

Fig.4. Fault Injection and Verification

Depending on the fault duration, different reconfiguration techniques are selected normal partial reconfiguration for a temporary fault or tiling technique for a permanent fault.



```
/fe/t1   01     01    00                01          00          01
/fe/t2   01     01                            00                            01
/fe/t3   01     01    00    01    10          01          10
/fe/t6   00     00                01    00          01    10
/fe/t7   00     00                                        01
/fe/q    0011   0011  0100  0101  0110  0111  1000  1001  1010  1011
/fe/t    0010   0010  0011  0100  0101  0110  0111  1000  1001  1010
/fe/x    0011   0011  0010  0110  0111  0101  0100  1100  1101  1111
```

Fig.5. Fault Injection and Verification

In the X value is the value of input is automatically generate the single input vector change format. That's why they change input vector as 0011,0010,0110,0111. Thus these are the changing the value as referring the SIC vector.

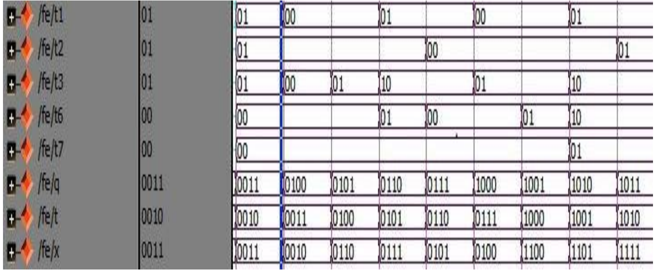
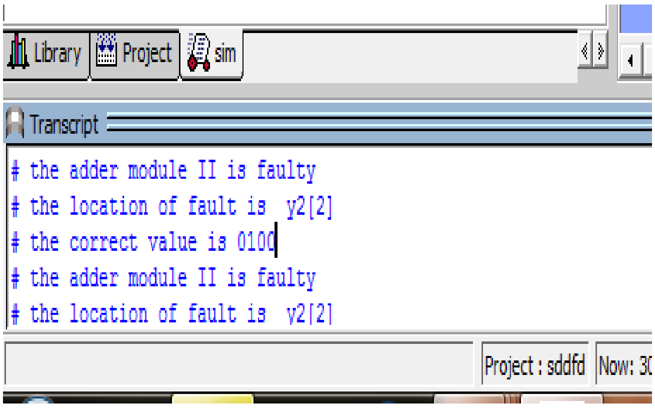

/fe/t1 01 | 01 00 01 00 01
/fe/t2 01 | 01 00 01
/fe/t3 01 | 01 00 01 10 01 10
/fe/t6 00 | 00 01 00 01 10
/fe/t7 00 | 00 01
/fe/q 0011 | 0011 0100 0101 0110 0111 1000 1001 1010 1011
/fe/t 0010 | 0010 0011 0100 0101 0110 0111 1000 1001 1010
/fe/x 0011 | 0011 0010 0110 0111 0101 0100 1100 1101 1111

Fig.6. Intermediate value and X value



```
# the adder module II is faulty
# the location of fault is  y2[2]
# the correct value is 0100
# the adder module II is faulty
# the location of fault is  y2[2]
```

Fig.7. Faulty Unit Identification Display

V. CONCLUSION

In this paper, we have proposed a new architecture of a fault-tolerant low cost system which can be implemented on FPGA with SIC. It was proposed and implemented on Xilinx Virtex-5 FPGA. Unlike the basic lockstep scheme, ours allows to identify the faulty core using a Configuration Engine which allows recovering from single-event upsets through partial reconfiguration combined with roll-forward recovery technique. As a result, the problem of fault latency is alleviated, because faults are detected immediately, once they cause an error. The scheme proposed is a valuable alternative to other fault.

REFERENCES

[1] H.Asadi,V.Sridharan, M.B.Tahoori, and D.Kaeli,(2005) "Balancing performance and reliability in the memory hierarchy," in Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw., vol. 25, no. 4 p.p 269–279.

[2] M. Ebrahimi, S.G.Miremadi, and H.Asadi,(2011) "ScTMR: A scan chain based error recovery technique for TMR systems in safety-critical applications," in Proc. Design Autom. Test Eur. Conf. Exhibit., vol. 36 no.3pp.1-4.

[3] M.Fazeli, A.Patooghy, S.G.Miremadi, and A.Ejlali,(2007) "Feedback redundancy: A power-aware SEU-tolerant latch design in DSM technologies," in Proc. 37th Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw., vol. 15 no. 4 pp. 276–285.

[4] J. M.Johnson and M.J.Wirthlin,(2010) "Voter insertion algorithms for FPGA designs using triple modular redundancy," in Proc. 18th Annu. ACM/SIGDA Int. Symp. Field Program. Gate Arrays, vol. 22, no. 5 pp. 249–258.

[5] F. L.Kastensmidt, L.Sterpone, L.Carro, and M.S.Reorda,(2005) "On the optimal design of triple modular redundancy logic for SRAM-based FPGAs," in Proc. Design Automatic Test Eur. Conf. Exhibit., vol. 18 no. 5p.p. 1530–1591.

[6] R.Leveugle, A.Calvez, P.Maistri, and P.Vanhauwaert, (2009,)"Statistical fault injection: Quantified error and confidence," in Proc. Design Autom. Test Eur. Conf. Exhibit., vol. 39 no. 5 pp. 502–506.

[7] A.Mohammadi, M.Ebrahimi, A.Ejlali, and S.G.Miremadi, (2012) "SCFIT: A FPGA-based fault injection technique for SEU fault model," in Proc. Design Autom. Test Eur. Conf. vol. 44, no. 4 pp. 586–589.

[8] S.Nomura, M.D.Sinclair, C.H.Ho, V.Govindaraju, M.de Kruijf, and K. Sankaralingam,(2011) "Sampling + DMR: Practical and low-overhead permanent fault detection," in Proc. 38th Int. Symp. Comput. Arch. vol. 47 no. 4 ,pp. 201–212.

[9] R.R.Rao, D.Blaauw, and D.Sylvester,(2006) "Soft error reduction in combinational logic using gate resizing and flip flop selection," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design, vol. 31 no. 4 pp. 502–509.

[10] G. Rui, C.Wei, L.Fang, D.Kui, and W.Zhiying,(2006,) "Modified triple modular redundancy structure based on asynchronous circuit technique," in Proc. 38th Int. Symp. Defect Fault Tolerance VLSI Syst., vol. 11 no. 3pp. 184–196.