A New Design Approach for Tolerating Interconnect Faults in Digital Circuits

R.V.Kshirsagar, *Fellow , IETE* and R.M. Patrikar, *Senior Member,IEEE*

Abstract--**This paper proposes an efficient new design approach for testing , detecting and tolerating single stuck-at-faults(s-a-0,s-a-1) at interconnect levels at the outputs of any digital circuit under test(CUT). The design is suitable to be used for highly dependable systems implemented by means of Field Programmable Gate Arrays(FPGAs) . Compared to alternate conventional designs, the one presented here allows achieving fault tolerance at lower design costs. All possible interconnect faults for wiring channels are considered. Signal routing in the presence of faulty interconnect resources is analyzed at both circuit level and the entire design level. This approach is demonstrated with the help of two full adders in which one is considered as the reference. In this approach we have an advantage of testing exhaustively the reference block and with random testing we can easily verify the functioning of other blocks so that time consumed to test other blocks gets reduced. In case of occurrence of stuck-at-faults at any interconnect, the circuit will reconfigure itself to select the other fault free output available on another interconnect , e.g., if s_1/c_1 (of fa_1) is stuck at any fault then circuit will choose s_2 /c_2 (of fa_2) as the output which is fault free.**

All possible interconnect faults are tested by injecting the faults using D - flip flop.

Index Terms-- **Ex-or gate, field programmable gate arrays (FPGA), fault tolerance, fault injection, reconfiguration, triple modular redundancy (TMR).**

I. INTRODUCTION

RELIABILITY and performance are the two important factors becoming major concern for next generation very deep sub-micron systems. Their reduced voltage supplies and therefore noise margins , together with their reduced internal capacitances, will dramatically increase their susceptibility and sensitivity to radiations and noise in general, making systems' failures extremely likely[7],[8]. As a consequence, not only systems oriented to mission critical applications (e.g., space, avionic, transport, etc.) will reinforce the use of fault-tolerance, but also general purpose systems implemented by next generation very deep sub-micron technologies, including FPGAs , will require the use of some form of fault tolerance[2],[3].

FPGAs are digital device that helps in implementing logic circuits by programming the required function; offer the benefit of low cost, short manufacturing turnaround time and easy design changes. As a result most prototypes and much production designs are now implemented on FPGAs, making hardware implementation economically feasible even for those applications which were previously restricted to software implementation.

To tolerate permanent faults in system hardware, redundancy is the most commonly used approach. Traditionally hardware redundancy is realized in a coarse grained level. This however is expensive in terms of area overhead. The regular structure of FPGAs allows remapping of logic from faulty areas to previously unused functional areas. This inherent ability of an FPGA to reconfigure itself at a fine grained level makes it ideal for fault tolerant implementations. The recent introduction of partial reconfigurable and run time reconfigurability in FPGAs has only served to aid efforts in this direction [1].

The common form of modular redundancy in practical systems is the Triple-Modular Redundancy (TMR) used for single event upset (SEU) mitigation. According to this technique, the considered basic block is triplicated and a majority voting circuitry (simply referred to as voter) is connected to the replicated blocks' outputs to give, at its output, the value present on the majority of its inputs,as shown in figure 1(a).The most common example of TMR is a d-type flip-flop that has been triplicated and to which a voter has been added on its output. By replacing all flip-flops in design with the circuit shown in figure 1(b), one would protect the design against SEUs in the flip-flops. However, this would not protect against SEUs in the combinatorial logic connecting the flip-flops in the design [5],[10],[11].The voter circuit is implemented using SRAM-cells which themselves are highly susceptible to upsets.

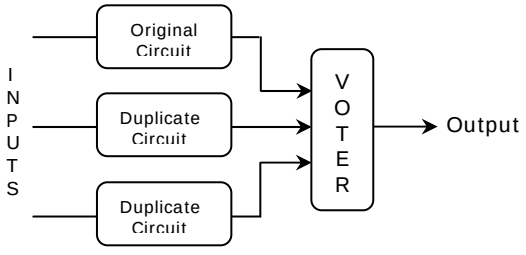

Fig. 1(a). Triple Modular Redundancy with voter

R. V. Kshirsagar is Professor with the Department of Electronics Engineering, Priyadarshini College of Engineering, Nagpur. He is pursuing his Ph.D. at Department of Electronics and Computer Science Engineering, Visvesvaraya National Institute of Technology, Nagpur, INDIA (e-mail: ravi_kshirsagar@yahoo.com).

R. M. Patrikar is Professor with the Department of Electronics and Computer Science Engineering, Visvesvaraya National Institute of Technology, Nagpur, INDIA (e-mail: rajendra@computer.org).

The effects of SEUs are not confined to the registers in digital designs, but are also present in the combinatorial logic for which there are several protection schemes proposed. These schemes mostly deal with transient glitches in the combinatorial logic that could result in upsets in the sequential elements.

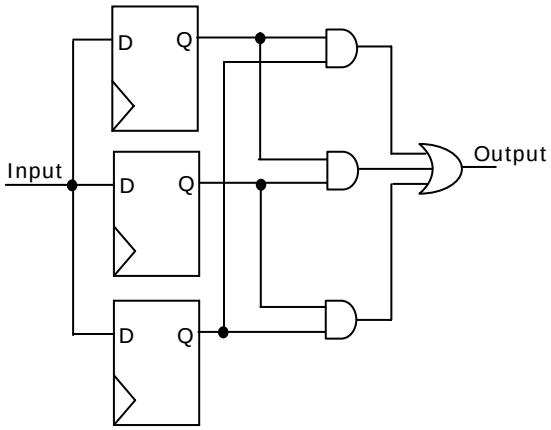

Fig. 1(b). Basic TMR circuit with d-ff and voter

II. METHODOLOGY

This paper presents a novel method to tolerate stuck at faults at interconnect levels at the outputs of any digital circuit under test (CUT). If there is a fault at one of the interconnects then circuit itself defines/detects the fault and configures to provide the fault free output. In place of the voter circuit we have used a novel circuit, as shown in figure 2, consisting of ex-or gates, priority encoders and multiplexers to produce fault free output at any moment of time. This approach allows achieving fault tolerance with respect to all possible interconnect faults.

The idea is presented with the help of two full adders and it can be extended to any desired level for any circuit.

The circuit under test (full adder here) is triplicated , out of which one is considered as the reference circuit(fa_r).The similar outputs of the CUT and its copies are then fed to the ex-or gates and compared with the similar outputs of the reference circuit. Again the similar outputs of the ex-or gates are fed to the priority encoders .The outputs of encoders are fed to two different multiplexers as select lines. The inputs to these multiplexers are the similar outputs of the two full adders, i.e., fa_1 and fa_2.

The circuit is designed with an assumption that the output of reference adder fa_r is always true as it has undergone through exhaustive testing by applying all the possible combinations of input test vectors and then by pseudo-random testing the working of other full adders can be verified for all possible interconnect stuck-at-faults(s-a-0, s-a-1).It is also assumed that only one fault occurs at a time. Pseudo-random testing requires less time which will be more helpful if the circuit has large number of inputs [6]. All possible interconnect faults are tested by injecting the stuck at faults using d-ff , although VHDL code can also be used for injecting and verifying the faults. The d-ff can be inserted at the output net of the circuit to be tested. The flip-flop is facilitated with reset and set inputs.

This technique can be generalized and implemented for tesing / tolerating faults in any other circuit.

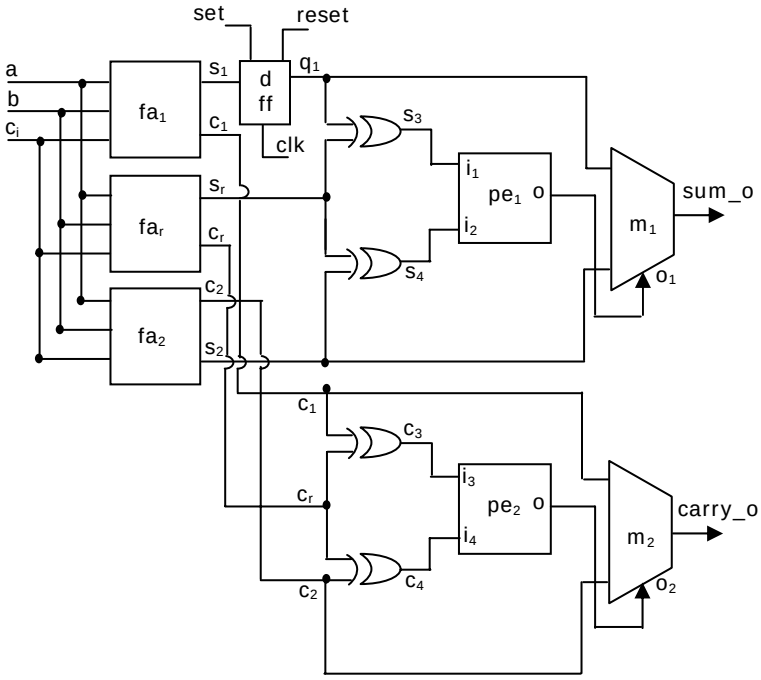

Fig. 2. Circuit to tolerate interconnect stuck-at-faults

III. TESTING STRATEGY

Three bits input is applied to the reference full adder and its copies. Outputs of full adders (fa_1 and fa_2) are s_1/s_2 and c_1/c_2, i.e., sum and carry respectively. Similarly the outputs of the reference full adder (fa_r) are s_r and c_r , i.e., sum and carry respectively. This system is designed in such a way that it tests the outputs of full adders (fa_1 and fa_2) and if any of them is stuck at any fault level then the circuit selects the fault free output through priority encoder which is finally propagated to the output (sum_o / carry_o) through mux_1 / mux_2.The truth table of full adder (exhaustive testing) is shown as below in TABLE I:

TABLE I
TRUTH TABLE (EXHAUSTIVE TESTING)

Input			Output	
a	b	c_i	s	c_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The circuit can also be tested for any stuck at fault at interconnect level using pseudo- random testing (TABLE II).

TABLE II
TRUTH TABLE (PSEUDO-RANDOM TESTING)

Input			Output	
a	b	c_i	s	c_o
0	0	0	0	0
0	1	1	0	1
0	0	1	1	0
1	1	1	1	1

If the input (abc_i) is "001" then outputs s_1 , s_2 and s_r are equal to '1' , and since both the inputs to the ex-or gates are same , the outputs s_3 and s_4 of ex-or gates will be '0' which are connected to the inputs of priority encoder .The priority encoder is designed in such a way that whenever i_1/i_3 is '0' then the outputs of priority encoders will be assigned a value '0' and whenever i_2/i_4 is '0' then the outputs of priority encoders will be assigned a value '1' , which are connected to the select lines(o_1/o_2) of mux_1/mux_2, hence s_1('1')/c_1('0') is propagated to the output sum_o /carry_o through mux_1/mux_2 , which is the correct (fault free) output. Here the priority is given to i_1/i_3 inputs. Now assume that s_1 is s-a-0 (i.e. d-ff is reset to '0'), then the output s_3 of ex-or gate becomes '1' (as $s_1=q_1$='0' and $s_r=s_2$='1'), i2 becomes '0', therefore o_1 will be '1' which is the select signal for mux_1 and hence s_2 ('1') is propagated to the output sum_o through mux_1 making it the fault free/correct output.

Now let us consider the input as "011".In this case outputs s_1 , s_2 and s_r will be '0', outputs s_3 and s_4 of ex-or gates will also be '0' and as i_1 is '0' ,the select line of mux_1 will be '0' making the final sum output (sum_o)as '0' . Now assume that s_1 is s-a-1 (i.e.d-ff is set to' 1') ,then the output s_3 of ex-or gate becomes '1' (as $s_1=q_1$='1' and $s_r=s_2$='0'),i_2 becomes '0' , therefore o_1 will be '1' which is the select signal for mux_1 and hence s_2('0') is propagated to the output sum_o through mux_1 making it the fault free/correct output.

Similar testing and detecting technique is applied using the remaining test vectors, i.e., "000" and "111" and by inserting the faults at either c_1 or c_2 for checking the carry output (carry-o) at the output of mux_2.

This design is also capable of producing the fault free outputs even in case of occurrence of any stuck at faults at the inputs of priority encoders / multiplexers and at the output of priority encoders. The only limitation of this circuit is that it can not detect and tolerate the faults at the input level of full adder (CUT) and the output of multiplexers.

IV. SIMULATION RESULTS

The fault-tolerance ability of the design has been tested and verified by means of VHDL simulation programs. We have written VHDL code for top entity and various components used in the design. We have used Modelsim SE 6.2e for verifying the design.

The simulation results are shown in figure 3 as follows :

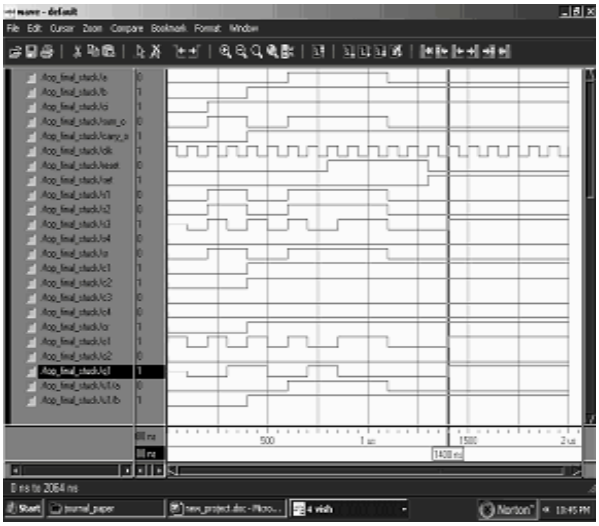

Fig. 3. Output waveforms for the top entity

V . SYNTHESIS RESULTS

Synthesis was done using the Xilinxs' synthesizer tool (XST) of ISE Foundation series 8.2i .Following are the results of synthesis :

Final Report

Final Results

RTL Top Level o/p File Name : top_final_stuck.ngr
Top Level o/p File Name : top_final_stuck
Output Format : NGC
Optimization Goal : Speed
Keep Hierarchy : NO

Design Statistics

IOs : 8

Cell Usage:

BELS : 2
LUT3 : 2
IO Buffers : 5
IBUF : 3
OBUF : 2

Device utilization summary:

Selected Device : v50pq240-5
Number of Slices : 1 out of 768 0%
Number of 4 input LUTs : 2 out of 1536 0%
Number of IOs : 8
Number of bonded IOBs : 5 out of 170 2%

Following is the RTL schematic of the top entity:

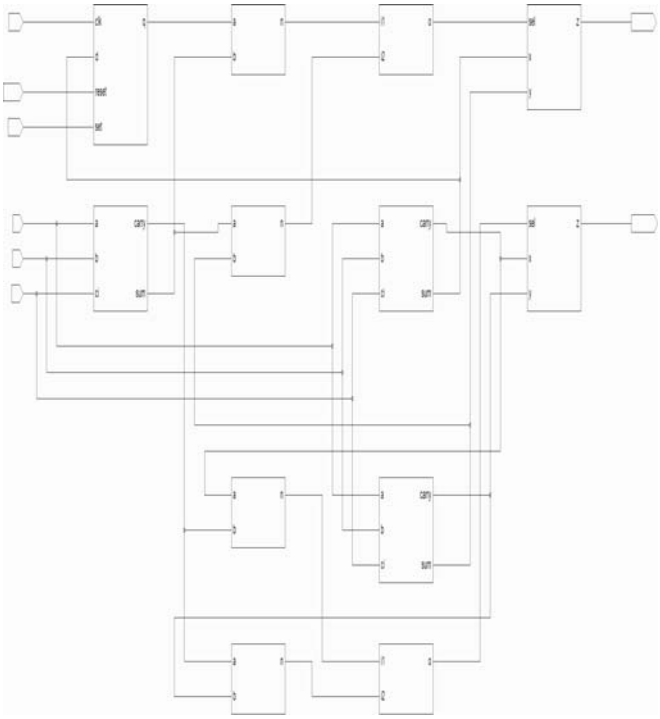

Fig. 4. RTL schematic

We have inserted the d-ff in the main design (figure 1a) for the purpose of only introducing the faults (‘0’and ‘1’) and testing the design. But in actual design the d-ff will not be there.

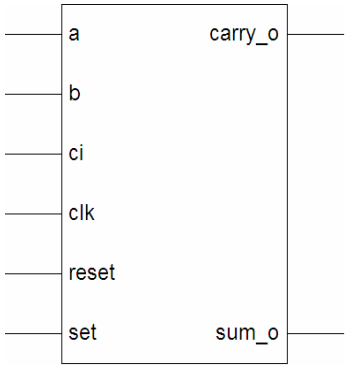

Fig. 5. Top Entity

VI. CONCLUSION

Many techniques have been suggested in past to detect and tolerate interconnect stuck at faults [4]. The technique presented in this paper discusses the stuck at faults for a novel circuit, which can be generalized for other such circuits also in the similar way [6]. We are also working on an algorithm which will enable the designer to inject the faults at VHDL level and test the circuit [9]. We have successfully implemented this design onto Xilinxs’ FPGAXCV50PQ240-5.

VII. REFERENCES

1. R. V. Kshirsagar, R .M. Patrikar, “Design of a Reconfigurable Multiprocessor Core for Higher Performance and Reliability of Embedded Systems”, IEEE Proc. on IFIP 14th Int Conf on Very Large Scale Integration, VLSI-SoC’06 , pp.251-254 , Oct.16-18 , 2006.
2. Monica Alderighi’, Sergio D’Angelol, Cecilia Metra’, and Giacomo R.Sechi “Novel Fault-Tolerant Adder Design for FPGA - Based Systems” , IIEEE Proceedings on On–Line Testing Workshop, 2001, Volume 7, 2001, pp. 54 -58
3. J.Lach and W. H. Mangione-Smith and M. Potkonjak, “Low Overhead Fault-Tolerant FPGA Systems”, IEEE Trans. on VLSI Systems, 1998, 6(2), pp. 212 - 221, June
4. F.Hanchek and S. Dutt, “Methodologies for Tolerating Cell and Interconnect Faults in FPGAs”, IEEE Transactions on Computers, 1998, Vol. 47, pp. 15 - 33, January
5. Sandi Habinc, “Functional Triple Modular Redundancy (FTMR) ” , Design and Assessment Report, Gaisler Research, FPGA- 003-01,ver.0.2, 2002,pp. 1-55, December
6. R. V. Kshirsagar, R. M. Patrikar etal, “Techniques for Fault Tolerance in FPGA”, Proc. on Emerging Trends in Engineering and Technology (FRONTIER-2007), 2007.
7. M.K. Stojcev, G.Lj. Djordjevic, T.R.Stankovic, “Implementation of self- checking two-level combinational logic on FPGA and CPLD circuits”, Journal of Microelectronics Reliability, issue 44, 2004, pp. 173-178
8. Lala PK, “Self checking and fault-tolerant digital system design” , San Fransisco: Morgan Kuffman Publisher, 2001
9. S.R.Steward, P.K.Lala, “Fault injection for Verifying Testability at the VHDL Level”, IEEE Proc. on ITC International Test Conference , 2003, pp. 131-137
10. F.Lima Kastensmidt, L.Sterpone, L.Carro, M.Sonza Reorda , “On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs”, IEEE Pro. on Design,Automation and Test in Europe Conference and Exhibition (DATE’05),2005
11. Fernanda Lima1,Luigi Carro, Ricardo Reis, “Designing Fault Tolerant Systems into SRAM-based FPGAs”, Proc. on DAC’03, June 2-6, 2003, Anaheim, California, USA,pp.650-655

VII. BIOGRAPHIES

Dr. Rajendra Patrikar (Sr.M,IEEE) is working presently as Professor in Electronics and Computer Science Department VNIT Nagpur. He joined asResearch engineer in the Microelectronics Project at IIT Bombay. He completed his Ph.D from the same department. He joined as faculty at IIT Bombay after working for a year at Computervision R&D Pune. Later he moved to Singapore to work in TECH Semiconductor in Advance Device Technology Department. After working there for three years he moved to Institute of High Performance Computing Singapore where he carried research work in the area of CAD for VLSI and nanoelectronics. He has published about 45 papers in International Journals, international conferences and also filed one patent in USA in the area of VLSI.

Ravindra V. Kshirsagar (FIETE,LMISTE) is presently working as Professor and Head of the Department of Electronics Engineering , Priyadarshini College of Engineering.He is also the chairman of Board of Studies(Electronics Engg.) of R.T.M.,N.U., Nagpur.
He has done his B.E.(E&TC) in 1984 from Govt. Engg. College, Jabalpur.He completed his M.Tech. (Electronics Engg.) in 1989 from VNIT,Nagpur.Currently he is pursuing his Ph.D. at VNIT,Nagpur. He has a vast teaching experience of 20 years and 2 years of industry experience. He has published many research papers in national and international conferences. He is a fellow member of IETE and LMISTE .Also he was Ex - IEEE member.
His special field of interest includes Reconfigurable Computing, VLSI Design, Fault tolerance and DFT.