Reliability Oriented Selective Triple Modular Redundancy for SRAM-Based FPGAs

Meisong Zheng^{1, a}, Zilong Wang^{2, b}, Ji Tu^{3, c}, Junye Wang^{4, d} and Lijian Li^{5,e} ^{1,2,3,4,5}Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100190 ^ameisong.zheng@ia.ac.cn, ^bzilong.wang@ia.ac.cn, ^cji.tu@ia.ac.cn, ^djunye.wang@ia.ac.cn, ^elijian.li@ia.ac.cn

Keywords: field programmable gate array (FPGA), triple modular redundancy (TMR), don't care bits (DC), windowing, single event upset (SEU).

Abstract. This paper presents an improved approach to *Triple Modular Redundancy (TMR)* which concerns *don' t care bits* of LUT configuration bits and hence classifies the set of LUTs into *SEU-sensitive* and *SEU-insensitive*. Unlike the full TMR approach, the improved approach only triplicates SEU-sensitive LUTs and can greatly reduces the area overhead while maintaining the circuit reliability. The proposed approach is thoroughly tested on the MCNC'91 benchmarks. Compare with the full TMR method the proposed scheme can reduce the area overhead by 26.6% on average, at the same time the circuit reliability only reduced by 9.1%. The improved approach can also increase *mean time between failures (MTBF)* by an average of six times more than the original circuit.

Introduction

SRAM based FPGAs are being increasingly used for modern digital systems since they offer low cost, reconfigurability, and low design turn-around time. But compared with ASICs, SRAM based FPGAs are more vulnerable to single-event upsets (SEUs) and as device size shrinks to the nanometer range this problem is increasingly disturbing.

The mitigation of single-event upsets in FPGAs is an increasingly important subject as FPGAs are used in radiation environments such as space. Triple Modular Redundancy (TMR) is the most commonly used fault tolerance technology by far, it was proposed as early as in the middle of the last century^[1,2]. Since TMR technology can tolerant only one module error, it is always implemented with the technology of configuration scrubbing ^[3, 4]. TMR together with scrubbing can effectively guarantee the FPGA system operation correctly, but the hardened design has 200% more area than the original circuit which is an intolerable area overhead for some design. In order to solve the above problems, a technique known as Selective Triple Modular Redundancy (STMR) based on sensitive gates definition has been proposed ^[5]. Since the STMR technique operates on a gate-level circuit, it is not suited to LUT network, so soon afterwards the sensitive ideas were transplanted into LUT level as Reduced Triple Modular Redundancy (RTMR) ^[6]. But as the authors described, because of fixing a threshold probability in order to determine the sensitivity of the LUTs is very difficult the RTMR technique does not proven to be effective.

This paper proposed a *don't care bits (DC)* based Selective Triple Modular Redundancy (bbSTMR) algorithm, which aims at reducing the area overhead of TMR system while maintaining roughly comparable fault tolerance capability. According to the statistical results of [7], $15\% \sim 40\%$ of the configuration bits in LUTs has the characteristics of *controllable don't cares*, consider of the propagation characteristics of the circuit, the *complete don't care bits* of some special circuit can even reach $60\%^{[8]}$. Experiments show that those don't care bits are mostly concentrate in some certain LUTs, on the benefits of this phenomenon, those LUTs boasts more don't care bits are regard as *SEU-insensitive* LUTs while the remaining are *SEU-sensitive*.

The core algorithm finds out as many as possible don't care bits for each LUTs using the technique of *windowing* at first, then SEU-sensitive and SEU-insensitive LUTs are defined by the number of

don't care bits and only the SEU-sensitive LUTs needs to be triplicate. In order to validate the performance of the algorithm, SEU fault simulations are implemented on the selective triplicate circuits. By implementation on several MCNC'91 benchmark circuits, the bbSTMR technique is shown to decrease the area overhead of full TMR circuits while maintaining a considerable reliability.

The remainder of this paper is organized as follows: Section 2 provide the proposed don't care bits based selective TMR algorithm. Section 3 gives the experimental results and the paper is concluded in Section 4.

Proposed bbSTMR Approach

Informal Overview. The proposed bbSTMR is a technique that selectively triplicates a circuit mapped while maintaining the fault tolerance capability of full triplicate circuits. This paper takes into account the criticality of LUT configuration bits only, interconnect configuration bits are not in the scope of consideration of this paper temporarily, and the size of LUT is 4-input hence each LUT have 16 configuration bits.

First, the Boolean circuit after mapping is described as a Directed Acyclic Graph (DAG) with LUTs as nodes and interconnects as edges^[9]; second, the complete don't care bits for all the nodes are calculated with the method of windowing, afterwards a database is established to store the number of don't care bits of each node; third, on the benefits of don't care bits statistics of every node, which LUTs are to be triplicate can easily determined; In the end, a SEU fault simulator is designed to verify the bbSTMR technique.

Don't care bits. For a given circuit y = F(x), Don't care bits (DC) as those configuration bits will not affect the normal function of the circuit when they are changed by SEU. If a configuration bit *b* is don't care, it should satisfy the formula as follows:

$$F_b(X) \equiv F_{\overline{b}}(X) | b \xrightarrow{SEU} \rightarrow \overline{b} . \tag{1}$$

Where X means to traverse the full input vector space of the circuit. For a LUT, the full set of DCs in the configuration bits should be found, and the number of DCs for the *ith* LUT is denoted as N_i .





The concept of don't care bits (DC) consists of controllable don't care bits (CDC) and observable don't care bits (ODC), whereby CDCs are configuration bits that can't be sensitized by any input and ODCs are those can be propagated to any output. An example of CDC and ODC is shown in Fig.1. As described in Fig.1(a), because one of the inputs of LUT z2 and LUT z3 fanouts from z1, they will never export an output of $z2 = 1 \land z3 = 0$, hence the configuration bit 10 in LUT F is uncontrollable, namely CDC. In Fig.1(b), when the inputs $X3 = 1 \land X4 = 1$, LUT z2 and LUT z3 will export a constant output of $z2 = 1 \land z3 = 1$ whatever the output of LUT z1 is, hence all of the configuration bits in LUT z1 is unpropagable with the inputs $X3 = 1 \land X4 = 1$, namely ODC.

Windowing and Sensitive LUTs definition. To find the exact set of don't care bits, the full input vector space of the circuit should be traversed. As the computational complexity grows exponentially with the number of input, it is impossible for large circuit.

The proposed bbSTMR approach utilize the windowing technique in [9] to calculate the number of don't care bits in each LUT, as described in [8] the windowing technique can guarantee a DC lower bound. Different with gate level circuit, almost every LUT has 4 inputs and the window expands badly as the transitive fanout level enlarged, so only one level of fanins and one level of fanouts for each node is concerned at present. And results show that even with only one level window, it is enough to obtain a large number of DCs.

Statistic data of the numbers of DCs in every LUT for some large circuit from MCNC'91 are provided in this paper. Table 1 shows the number of LUTs of 0~16 DCs for every circuit and the total number of LUTs are listed at the bottom. From table 1 there is an apparent phenomenon that the DCs concentrates on certain LUTs, for example in the circuit of alu2 there are 6 LUTs boasts 16 bits of DCs meaning that however these LUTs are affected by SEUs, errors will not appear at the circuit output. According to this phenomenon the set of SEU-sensitive and SEU-insensitive can be defined.

Table 1 Statistics of numbers of DCs											
NCB	alu2	ex5p	misex3	des	seq	apex2	spla				
0	93	525	675	1121	966	1113	1859				
1	0	0	0	0	0	0	0				
2	4	11	12	1	4	18	92				
3	2	6	0	1	1	0	12				
4	4	62	69	84	66	43	284				
5	1	18	14	5	9	4	45				
6	1	24	11	61	16	13	60				
7	8	40	16	13	20	13	143				
8	41	154	387	260	498	524	843				
9	3	16	23	57	14	8	59				
10	2	41	21	37	11	1	105				
11	2	25	5	6	2	3	17				
12	23	69	131	83	155	122	94				
13	2	16	6	3	3	5	13				
14	2	21	29	58	17	10	28				
15	3	9	2	0	3	0	20				
16	6	28	10	57	6	1	16				
total	197	1065	1397	1591	1750	1878	3690				

LUT with more don't care bits means that it is more immune to SEU, so it is a SEU-insensitive LUT. To define the sensitive and insensitive LUTs, a threshold value H is introduced. If the number of DCs N_i in a certain LUT_i Satisfies the condition N_i <H then LUT_i is SEU-sensitive, otherwise LUT_i is SEU-insensitive. For the 4-input LUT model this paper, we define H = 8, and experimental results show that this value is suitable for the selective TMR design.

SEU Simulation. In order to verify the reliability of proposed method, a SEU fault simulator is designed. The simulator implements fault injection, circuit simulation, fault statistics and provides a comprehensive evaluation for the circuit. A single fault simulation process works as follows:

- 1. Randomly choose one node in the circuit to be SEU affected;
- 2. Randomly generates an input vector;
- 3. Perform logic simulation and record the output values;
- 4. Invert the value of the chosen node in step 1;
- 5. Perform logic simulation and compare the output values with the result of step 3;
- 6. Update the fault information of the circuit.

Experimental results

To evaluate the SEU immunity and the area overhead of the proposed bbSTMRed circuit, the proposed bbSTMR method is thoroughly tested on MCNC'91 benchmarks. The reliability evaluation is performed by randomly inject 1000 faults for each circuit and the results is shown in Table 2. The bbSTMR technique performs better than [5] in SEU sensitivity and can match the results in [6]. On average the bbSTMR technique requires only 120% additional redundancy compared to the 200% requirement of full TMR while maintaining a comparable SEU immunity with the full TMR. The failure rate and mean time between failures (MTBF) are also calculated for each circuit and the ratio is calculated by dividing the MTBF of bbSTMR circuit by the MTBF of the original circuit. As the results listed in the last column of Table 2, the MTBF of bbSTMR circuit increased six times than original circuit on average.

Name	LUTs No.in ori.	No. of faults	LUTs No.in bbSTMR	No. of faults	% of extra LUTs	LUTs No. in TMR	% of saved LUTs	reli. of ori.	reli. of bbSTMR	MTBF ratio
c1355	74	654	206	12	178.38	222	7.21	34.6	98.8	19.6
c499	74	652	206	8	178.38	222	7.21	34.8	99.2	29.3
c432	124	240	284	28	129.03	372	23.66	76	97.2	3.74
c880	174	491	392	76	125.29	522	24.90	50.9	92.4	2.87
c8	39	694	69	217	76.92	117	41.03	30.6	78.3	1.81
сс	26	733	48	147	84.62	78	38.46	26.7	85.3	2.70
cm138a	10	695	14	279	40.00	30	53.33	30.5	72.1	1.78
cm152a	6	333	14	22	133.33	18	22.22	66.7	97.8	6.49
alu2	197	397	423	55	114.72	591	28.43	60.3	94.5	3.36
alu4	1522	268	3294	39	116.43	4566	27.86	73.2	96.1	3.18
apex2	1878	316	4286	29	128.22	5634	23.93	68.4	97.1	4.77
des	1591	724	3633	112	128.35	4773	23.88	27.6	88.8	2.83
miex3	1397	474	2903	98	107.80	4191	30.73	52.6	90.2	2.33
seq	1750	409	3774	65	115.66	5250	28.11	59.1	93.5	2.92
spla	3690	473	8680	66	135.23	11070	21.59	52.7	93.4	3.05
ex5p	1064	403	2434	42	128.76	3192	23.75	59.7	95.8	4.19
					120.07		26.64	50.28	91.91	5.93

Table 2 Experimental results of area overhead and SEU sensitivity

Conclusion and future work

In this paper, we have presented a bbSTMR method that can selectivly triples a circuit and the preliminary experiment results are encouraging. With the reliability of up to 92% and the six times increase in MTBF our method together with the confiugration scrubbing technique can ensure a SRAM based FPGA system operating steady.

Our future work will explore the windowing technique to impove the don't care bits calculation and shorten the calculation time. We can increase the transitive level of those windows with less leaves or limit the number of leaves by overlapping windows. Also, we would like to expand out work to K-input(K>4) LUT model.

References

- [1] Triple Module Redundancy Design Techniques for Virtex FPGAs[X], XAPP197(v1.0), Xilinx Corp., 2001.
- [2] Kretzschmar, et.al. Robustness of different TMR granularities in shared wishbone architectures on SRAM FPGA, 2012 International Conference on Reconfigurable Computing and FPGAs (ReConFig), p.1-6.
- [3] Melanie Berg, *et.al.* Effectiveness of Internal Versus External SEU Scrubbing Mitigation Strategies in a Xilinx FPGA: Design, Test, and Analysis, IEEE transactions on nuclear science, vol.55, Issue. 4, Aug. 2008: 2259~2266.

- [4] Correcting Single-Event Upsets in Virtex-II Platform FPGA Configuration Memory[X], xapp779(v1.1), Xilinx Corp., 2007
- [5] Samudrala,P.V., Ramos,J. Selective triple modular redundancy(STMR) based single-event upset(SEU) tolerant synthesis for FPGAs, IEEE transactions on nuclear science, vol.51, NO. 5, October 2004: 2957~2969.
- [6] Chandrasekha, V., Mahammad, S.N., Muralidharan, V. Reduced Triple Modular Redundancy for Tolerating SEUs in SRAM based FPGAs, 2005 MAPLD International Conference.
- [7] Zhe Feng, Naifeng Jing, Yu Hu, Lei He, IPF: In-place X-filling to mitigate soft errors in SRAM-based FPG As, in Proceeding the 21st International Conference on Field Programmable Logic and Applications (FPL), Sept. 2011, pp. 482-485.
- [8] Cong, J., Minkovich, K. LUT-based FPGA technology mapping for reliability. 47th ACM/IEEE Design Automation Conference (DAC), 2010, Page(s): 517 522.
- [9] Mishchenko, A. andBrayton, R. SAT-Based Complete Don't-Care Computation for Network Optimization, Proceedings of the Conference on Design, Automation and Test in Europe, pp. 412-417, Mar. 2005.