Fast SEU Detection and Correction in LUT Configuration Bits of SRAM-based FPGAs

Hamid R. Zarandi[1,2], Seyed Ghassem Miremadi[1], Costas Argyrides[2], Dhiraj K. Pradhan[2]

[1]*Department of Computer Engineering, Sharif University of Technology, Azadi Ave. Tehran, Iran*
[2]*Department of Computer Science, Bristol University, Bristol BS 1UB, UK*
zarandi@ce.sharif.edu, miremadi@sharif.edu, costas@cs.bris.ac.uk, pradhan@cs.bris.ac.uk

Abstract

FPGAs are an appealing solution for the space-based remote sensing applications. However, in a low-earth orbit, configuration bits of SRAM-based FPGAs are susceptible to single-event upsets (SEUs). In this paper, a new protected CLB and FPGA architecture are proposed which utilize error detection and correction codes to correct SEUs occurred in LUTs of the FPGA. The fault detection and correction is achieved using online or offline fast detection and correction cycles. In the latter, detection and correction is performed in predefined error-correction intervals. In both of them error detections and corrections of k-input LUTs are performed with a latency of 2^k clock cycle without any required reconfiguration and significant area overhead. The power and area analysis of the proposed techniques show that these methods are more efficient than the traditional schemes such as duplication with comparison and TMR circuit design in the FPGAs.

1. Introduction

SRAM-based field programmable gate arrays are being increasingly used to start new designs because of their growing density and speed, reconfigurability, shot-design cycle and cost-effectiveness [1]. While the use of reprogrammable FPGAs offers a number of important advantages, these SRAM-based FPGAs are very sensible to heavy ion, proton and neutron induced single event upsets (SEUs) [3], [6], [9].

There are many available resources within an FPGA to perform various logic functions. The way in which these resources are utilized and interconnected is specified by the circuit design, also known as a configuration bitstream. The configuration bitstream determines which resources within the FPGA are used to implement a specific logic design.

1-4244-0910-1/07/$20.00 c2007 IEEE.

The effect of the SEU on the configuration memory of an FPGA, would lead to a permanent error which remains in the FPGA until the next reconfiguration of a new design [4]. This permanent error may result in a logic error or routing error depending on which part of the configuration memory is affected. A logic error may lead to complement one of the entries of the Look-Up Tables (LUTs) modifying the functionality of the mapped logical function [8]. A routing error may lead to a signal getting misrouted or disconnected [5], [9].

Error detection and correction code (EDAC) is a well-known technique for protecting storage devices against transient faults [7]. An example of EDAC is the Hamming code, which is useful for protecting memories against SEU because of its efficient ability to correct single upsets per coded word with reduced area and performance overhead.

In this paper, we introduce three different schemes for detecting and correcting errors in configuration bits of the LUTs. These schemes can be applied at different level of FPGA structure: 1) FPGA-level, which every line of CLBs within the FPGA is protected, 2) CLB-level, which protection is performed for every CLB (a set of LUTs) and 3) LUT-level, which protection is performed for every LUT. In these schemes, the error detection and correction is provided in one clock cycle and is independent to the number of CLBs located in SRAM-based FPGAs.

The experimental studies show that using the proposed schemes in FPGAs, all single and double SEUs are detectable and single SEUs is correctable in just one clock cycle without any FPGA reconfiguration and is independent to the number of CLBs. Moreover, using the proposed schemes, the area and power overhead of the new circuit design is more efficient than the previous schemes such as duplication with comparison (DWC) [11].

The rest of this paper is organized as follows. Section 2 presents some related work. Section 3 introduces the protection code and the proposed schemes for the FPGAs. The CLB architecture for fast detection and correction is presented in section 4. Section 5 calculates the probability of having multiple uncorrectable errors in protected Xilinx Viretex II FPGA family. Section 6 compares area, power and correction capability of the proposed technique with related work. Finally section 7 concludes the paper.

2. Related work

In order to overcome SEUs affecting the FPGA configuration memory, several fault-tolerance methods have been proposed in the past years. One of techniques, called scrubbing, is periodically reloading the whole content of the configuration memory [1]. By the use of readback and partial reconfiguration capabilities of FPGAs, a recovery system can be used [2]. Through the readback option, the content of the FPGA's configuration memory is read and compared with the expected one, which is stored in a predefined memory located outside of the FPGA. If a mismatch is found, the correct information is downloaded in the FPGA's configuration memory. During reconfiguration only the faulty portion of the configuration memory is overwritten. There are several fault-tolerant techniques that do not consider detection and correction occurred SEUs, but just aim at masking errors not to propagate elsewhere. These methods are proposed mainly by hardware redundancy.

Triple Modular Redundancy (TMR) is a well-known fault-tolerant technique for preventing error propagation [3]. The TMR implementation uses three identical logic blocks performing the same task in parallel regarding to outputs being compared through majority voter. However, this solution enforces high area overhead, three times more input and output pins, high performance penalties [6]. Moreover, it may not be affordable to put redundancy in each and every module (or component) especially in embedded systems where power and area are important constraints. Another error mitigation technique which is based on modular redundancy and time redundancy has been proposed in [11] which uses Duplication with Comparison (DWC) and Concurrent Error Detection (CED) to create a fault-tolerant system. However, this method is depended on the logic of the circuit that is mapped on to the FPGA and suitable encoding and decoding functions for each such block

3. Error detection and correction

The proposed approach is to use SEC-DED codes in several separated LUTs to detect and correct the errors in a cluster of LUTs rather than to detect one LUT as in the case of DWC technique. Employing SEC-DED codes in the FPGA can be done at different grains:

- To use protection codes in each horizontal (vertical) line of FPGA through CLBs (FPGA-level).
- To use protection codes in each CLB (CLB-level protection).
- To use protection codes in each LUT (LUT-level protection).

We define the ratio of dividing number of protection code bits used in the FPGA by the number of FPGA bits that are protected as protection-granularity.

The protection-granularity of the last case is greater than the other cases while the protection-granularity of the first case is less than the others. As the protection-granularity of a protected FPGA increases, the probability of being an error in configuration bits of FPGA decreases. However, the area overhead, and therefore the power of the protected FPGA would be increased. Hence, there is a trade off between the area (and power) overhead and the fault-tolerant capability to protect an FPGA, that designers of the FPGA-based circuits should determine.

Based on design and implement of mentioned cases, the second case with medium protection-granularity is suggested. In this scheme, for a cluster of N LUTs, the number of K LUTs are dedicated to check the errors of N LUTs and to correct the error in the case of single error in one of the N LUTs. Therefore, the following equation should be satisfied.

$$N+K+1 <= 2^K \quad (1)$$

This means that for a cluster of N LUTs, about $log(N)$ LUTs are needed for storing the protected codes and this overhead is very considerable with the DWC and TMR approaches which impose at least two and three times area and power overhead, respectively. It can be shown that, the bitwise protection coding is sufficient to detect and correct errors. Consider a line of 16 LUTs (N=16) that are protected by 5 extra LUTs (K=5), and each LUT maps m-input boolean function. Let D_i be the j^{th} bit of i^{th} LUTs. Therefore, the bitwise parity bits of the SEC-DED code is computed as follow.

$$P_0 = D_{13} \otimes D_{11} \otimes D_{10} \otimes D_9 \otimes D_6 \otimes D_5 \otimes D_3 \otimes D_0, \quad (2)$$

$$P_1 = D_{14} \otimes D_{13} \otimes D_{12} \otimes D_{11} \otimes D_{10} \otimes D_7 \otimes D_6 \otimes D_4 \otimes D_1, \quad (3)$$

$$P_2 = D_{15} \otimes D_{14} \otimes D_{10} \otimes D_9 \otimes D_8 \otimes D_7 \otimes D_6 \otimes D_3 \otimes D_2 \otimes D_0, \quad (4)$$

$$P_3 = D_{15} \otimes D_{11} \otimes D_{10} \otimes D_9 \otimes D_8 \otimes D_7 \otimes D_4 \otimes D_3 \otimes D_1, \quad (5)$$

$$P_4 = D_{12} \otimes D_{11} \otimes D_{10} \otimes D_9 \otimes D_8 \otimes D_5 \otimes D_4 \otimes D_2, \quad (6)$$

where P_k is j^{th} bit of the k^{th} LUTs.

When the protected FPGA is programmed, these protected bits are computed and stored in the protection LUTs. During testing period the new protected bits are computed and compared with the original stored ones. The result of this comparison, call syndrome, will indicate the incorrect bit position in a single error. In this protection code, there is an overall parity that computed by all bits. Using the syndrome and overall parity comparison, fault detection and correction is available. Double bit errors are detected when syndrome is not zero but the overall parity is zero. In the case of single faults, the overall parity

comparison is not zero and the syndrome indicates the location of fault occurrence bit.

3.1. FPGA-level protection

Figure 1 shows a simple example of the implementation of FPGA-level protection in which the protection codes are considered for a row of FPGA with four columns FPGA and the protected FPGA's columns are increased to seven columns. The gray box show modifications needed to implement the protection code. In this scheme, the "Generate SEC-DED and Comparator" can be shared for all of the LEs inside of CLBs in a row of FPGA. At time of fault detection and correction, the contents of LUTs inside of one CLB row are read and the syndrome and overall parity are generated. Therefore, for testing each of LUTs, the detection and correction should be repeated 2^k times where k is the input number of a LUT. Assume that each CLB has M LUTs, a k-bit counter is necessary to be located inside of each CLB for addressing each bit position of LUTs in parallel, and therefore, the results of each LUT would be checked by the SEC-DED generation codes. In this scheme, some modification can be applied for decreasing the area overhead of FPGA-level protecting. As each LUT inside of CLBs are checked by corresponded SEC-DED circuitry, therefore we need M different SEC-DED circuitry for each LUTs where M is the number of LUTs inside of CLBs. However, we can use just one SEC-DED circuitry and share it for all rows of LUTs. This make the area overhead to be decreased but the testing time of each CLB will be increased. We considered these two schemes with name of FPGA-level with and without shared circuitry in the experimental results. In order to implement this level of FPGA protection, several modifications are necessary in the x-channel connections. There are several direct lines namely *direct*, *double*, *hex* and *long* lines, in X and Y channels in Xilinx FPGAs which connect two CLBs that are one, two, six and one row far apart, respectively. These given lines in Xilinx FPGAs can be utilized for implementing the connections required for the protection code circuitry. Although available X-channel connections can be utilized for the circuitry of producing SEC-DED codes, however this cause the flexibility of X-channels in routing circuits inside of FPGA to be decreased. Hence, for implementing this level of protection, embedding several direct wires between information bits and circuitry of SEC-DED is desirable.

It should be noted that in this scheme, the length of information bits which is used for protecting is based on the number of columns that FPGA has. This means that the protection capability of this scheme is significantly depended to FPGA size. For example, if the dimensions of FPGA increase, the protection capability of this scheme would decrease.

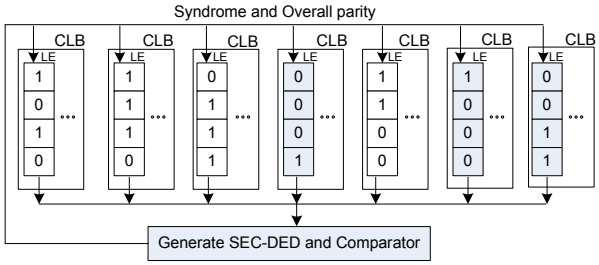

Figure 1. FPGA-Level protection: an FPGA with SEC-DED protection CLBs

3.2. CLB-level protection

Utilizing SEC-DED codes can be applied for all of LUTs inside of a CLB. Figure 2 shows a CLB which is protected by protection code. B0, B1, B3 are LUTs required for storing the protection codes of B2, B4, B5 and B6 LUTs. In this case, all bits in the same significant bit positions in different LUTs are protected in the same significant bit positions in the protection blocks.

In this architecture, since the information and protection bits are stored apart and in separated blocks (LUTs), therefore the probability of having more than double errors in each LUT of information and protection bits will be decreased significantly. In this case, all of multiple errors occurred in only one LUT of a CLB can be detected and corrected but if multiple errors occurred in different LUTs of a CLB in same bit positions, they may be detected providing that the number of errors is equal or less than two.

In order to implementing this level of protection, a k-bit counter is required to address different bit position of each LUTs. The detection and correction of errors in LUTs of a CLB can be achieved by 2^k times of detection and correction for each bit inside of a LUT. The main difference between this level of protection and FPGA-level one is that the information bits in this scheme are much less than the other one. Moreover, all connections between information and protection bits are router inside of CLB internally and therefore this method is more modular than the previous one. However, the area overhead of this scheme is more than FPGA-level. The implementation of CLB-level protection codes can be done in two different cases with and without sharing the SEC-DED circuitry. In the case of sharing SEC-DED circuitry, area overhead of protection is decreased but the time of detection and correction of errors will be increased 2^k times.

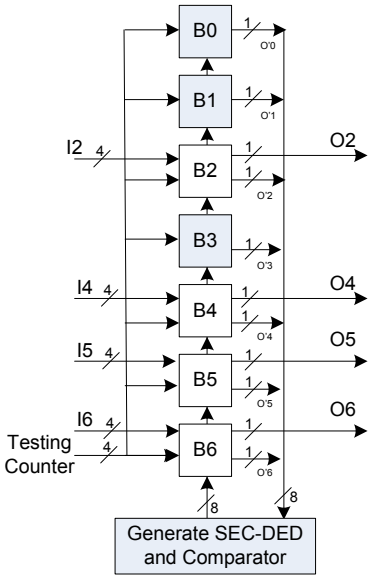

Figure 2. CLB-Level protection: a CLB with SEC-DED protection LUTs

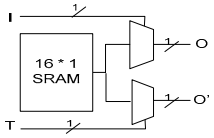

Figure 3. A dual-port 4-input LUT

In figure 2, each 16×1 LUT is replaced by a dual-read LUT shown in figure 3. Therefore, every CLB has 4 additional input lines that consist of the four output lines of testing counter. The testing counter is a 4-bit counter, 0-15 binary up-counter, provided either on FPGA chip or kept as a stand-lone counter, incremented once every clock cycle. In addition to the LUTs used by the circuit mapped to the FPGA, a few SEC-DED LUTs are also added to every CLB of FPGA. These SEC-DED LUTs store the pre-computed 16-bit SEC-DED check bits of the other LUTs of CLB. The architecture shown in figure 2 performs at-speed detection and correction of single error of configuration bits of LUTs without disturbing the normal functioning of the FPGA.

3.3. LUT-level protection

Figure 4 shows employing SEC-DED codes used in a LUT. The gray shapes in this figure show the modifications needed for implementing it in a LUT. In this scheme, each LUT in a FPGA has its own protection code and therefore all double errors inside of one LUT can be detected and all single errors inside of one LUT can be corrected. The area overhead of this scheme is more than the previous two schemes since each LUT has separated protection circuitry.

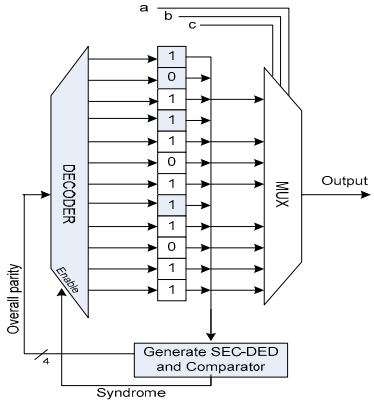

Figure 4. LUT-level protection: a LUT with SEC-DED protection bits

4. CLB architecture for detection and correction errors

Based on these three mentioned protection levels, CLB-level protection is suggested for employing in FPGAs. Figure 2 shows the proposed technique applied on CLB architecture. The main reasons are that CLB-level protection is less complex than FPGA-level to layout the FPGA by manufacturers since the protection structure is localized in each CLB architecture and the protection code routings are inside of each CLB. Typically, manufacturers manually layout a single tile consisting of logic block and switch block and replicate them across the entire chip. Therefore, CLB-level and LUT-level protection schemes are better ones for implementing compared to FPGA-level. However, CLB-level protection is more reliable than LUT-level, since the information bits that is protected by each check bits are distributed through several LUTs. So it can detect and correct multiple errors occurred in a LUT while in the LUT-level because of using protection codes for each LUT, only detecting double errors and correcting one error is achievable.

Without loss of generality, we assume that FPGA design used for the fault tolerance is composed of 16 × 16 CLBs arranged in a square matrix and each CLB consists of 8 3-input LUTs. Table 1 shows different implementation of the mentioned schemes and compare them in terms of information and check bits, area, delay of detecting and correcting information bits and delay of testing whole FPGA.

In this table, T_8 and T_{16} are time requited for performing detection and correction of single error in 8 and 16 bits, respectively. For each protection scheme, two cases of implementation are considered based on sharing or not sharing the hardware implementation for a group of similar information bits. For example, in the LUT-level with shared hardware, all LUTs inside one CLB are

considered to share the hardware needed for encoding, decoding logics needed for SEC-DED code. FPGA-level protection scheme has less area overhead compared to CLB-level and LUT-level. However, FPGA-level testing time is more than other cases because it protects information bits more than others. Moreover, it is more complex to implement compared to others. CLB-level and LUT-level have same area and delay overhead, since the size of LUT and number of LUTs inside one CLB is same in the mentioned FPGA. However, CLB-level is more powerful in terms of correction multiple faults in one LUT.

Table 1. Comparison of area and delay overheads and FPGA testing time for 16 × 16 FPGA (CLB size= 8 3-input LUTs)

Level of protection codes	# of information + check bits	area (μm^2)	delay (ns)	FPGA testing delay
FPGA-level (with shared HW)	16384+5120	3842304	12	64 T_{16}
FPGA-level (without shared HW)	16384+5120	4376576	12	T_{16}
CLB-level (with shared HW)	16384+8191	4225536	8	8 T_8
CLB-level (without shared HW)	16384+8191	6355742	8	T_8
LUT-level (with shared HW)	16384+8191	4116992	8	8 T_8
LUT-level (without shared HW)	16384+8191	6170624	8	T_8

5. Detection and Correction of multiple faults

Based on the different levels of protections, the CLB-level is the best level for the protection using SEC-DED method since it can detect multiple adjacent faults in the LUTs with good level of modularity and less complexity. Although the LUT-level can detect double errors and correct single errors in each LUT, however the cost of implementation of CLB-level is close to the LUT-level. We propose the CLB-level protection method to be used in the FPGA. In this case, any double faults in same significant bit position of LUTs of a CLB are detectable while single fault at each significant bit position of LUTs of a CLB is correctable. In this scheme, the probability of undetecting double errors inside of a FPGA will be decreased significantly.

When double configuration upsets occurred, if they happened in same bit positions of LUTs of a CLB cannot be correctable. Let N be the number of LUTs in the device and each CLB composed of 8 4-input LUTs and CLB-level protection is employed in the FPGA. The probability of two, three and four configuration upsets uncorrectable by this scheme are given by:

$$p_{2\,uncorrectable\,errors} = \frac{\binom{16N}{1}\cdot\binom{12}{2}}{\binom{192N}{2}}, \tag{6}$$

$$p_{3\,uncorrectable\,errors} = \frac{\binom{16N}{1}\cdot\binom{12}{3}+\binom{16N}{2}\cdot\binom{12}{2}\cdot\binom{12}{1}}{\binom{192N}{3}}, \tag{7}$$

$$p_{4\,uncorrectable\,errors} = \frac{\binom{16N}{1}\cdot\binom{12}{4}+\binom{16N}{2}\cdot\binom{12}{2}\cdot\binom{12}{2}+\binom{16N}{2}\cdot\binom{12}{1}\cdot\binom{12}{3}+\binom{16N}{3}\cdot\binom{12}{1}\cdot\binom{12}{1}\cdot\binom{12}{2}}{\binom{192N}{4}}, \tag{8}$$

We computed these probabilities for a series of Xilinx FPGAs which are mentioned in Table 2. As this table shows, the probability of having 2 uncorrectable errors for the mentioned scheme is very low and this probability decreases when the size of FPGA increased. However, the probability of having three or four uncorrectable in FPGA is more than the probability of having two uncorrectable in FPGA because the employed protection code can correct single fault. It should be noted that in the real application the probability of occurring three and four errors is considerably less than the probability of occurring two errors. Therefore, if the correction of LUT contents happened in appropriate time slots, the content of LUTs will not be erroneous.

6. Area and Power Comparison

The CLB architecture shown in figure 2, parity-protected, DWC and TMR FPGA architectures were synthesized with Synopsys® CAD tool and 0.18 micron CMOS technology to compare the area, power and delay requirements. The advantage and disadvantage of proposed architecture over standard DWC technique in terms of area, power, delay and additional configuration memory requirements are shown in Table 3. The area overhead of parity-protected CLB architecture is about 48 percent regarding to the area of simple CLB architecture. The area overhead of DWC and TMR methods are also about 79 and 204 percent compared to the simple CLB architecture. Based on these results the area overhead of our proposed technique is less than DWC and TMR schemes. In the case of power consumption, the parity-protected CLB architecture consumes less power among the other protection schemes, but it can be only used for detecting errors. However, the power consumption of the proposed technique is less than the DWC and TMR schemes. This is expected since the implementation of the proposed hardware causes several extra check bits and routes to perform error detection and correction.

Table 2. The probability of multiple errors not being correctable for protected Xilinx Virtex II FPGAs

Device	No. of CLBs	Prob. of having 2 uncorrectable errors	Prob. of having 3 uncorrectable errors	Prob. of having 4 uncorrectable errors
XC2V40	8 × 8	8.95E-004	1.30E-003	1.80E-003
XC2V80	16 × 8	4.48E-004	6.71E-004	8.95E-004
XC2V250	24 × 16	1.49E-004	2.28E-004	2.98E-004
XC2V500	32 × 24	7.45E-005	1.12E-004	1.49E-004
XC2V1000	40 × 32	4.47E-005	7.71E-005	8.95E-005
XC2V1500	48 × 40	2.98E-005	4.48E-005	5.97E-005
XC2V2000	56 × 48	2.13E-005	3.20E-005	4.26E-005
XC2V3000	64 × 56	1.60E-005	2.40E-005	3.20E-005
XC2V4000	80 × 72	9.95E-006	1.49E-005	1.98E-005
XC2V8000	112 × 104	4.92E-006	7.37E-006	9.83E-006

Table 3. Comparison of area, power and configuration memory requirement for a CLB

CLB architecture	No. of LUTs	Area		Power		No. of SRAM bits	Single Error Detection	Double Error Detection	Error Correction
		μm^2	%	µw	%				
Standard FPGA (Virtex II)	8	10240	100	230	100	128	0%	0%	0%
Protected FPGA with parity [7]	9	15258	149	331	144	144	100%	0%	0%
Duplication with comparison	16	16282	179	525	228	256	100%	100%	0%
TMR-based FPGA	24	31130	304	802	348	348	100%	100%	0%
Our proposed FPGA	12	16506	161	532	231	160	100%	100%	100%

7. Conclusions

In this paper, we have presented a new FPGA architecture which detect and correct errors of LUT configuration bits. This is achieved by employing SEC-DED codes in CLBs of FPGA architecture. Hence, three different implementation of SEC-DED based FPGA architecture were introduced and explored and the best one is proposed for implementing in FPGA. The analytical results have shown that using the proposed CLB architecture improves the reliability of CLB so that the probability of having two uncorrectable errors in a CLB is decreased significantly. The results of implementation comparison has shown that this method impose less area and power overhead compared to the previous fault-tolerant schemes such as duplication with comparison and triple modular redundancy schemes.

References

[1] "Correcting Single Event Upsets Through Virtex Partial Reconfiguration", Xilinx Application Note XAPP216, June 2000.

[2] M. Gokhale, P. Graham, E. Johnson, N. Rollins, M. Wirthlin, "Dynamic Reconfiguration for Management of Radiation-Induced Faults in FPGAs," *18th IEEE Parallel and Distribution Processing Symposium*, pages 145-150, 2004.

[3] F.L. Kastensmidt, L. Sterpone, L. Carro, M. Sonza Reorda, "On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs," *IEEE Design, Automation and Test in Europe*, pages 1290-1295, March 6-10, 2005.

[4] M. Sonza Reorda, L. Sterpone, M. Violante, "Multiple errors produced by single upsets in FPGA configuration memory: a possible solution," *IEEE European Test Symposium*, pages 136-141, 2005.

[5] E.S.S. Reddy, V. Chandrasekhar, M. Sashikanth, V. Kamakoti, "Detecting SEU-caused Routing Errors in SRAM-based FPGAs," *18th International Conference on VLSI Design*, pages 736-741, 2005.

[6] G. Asadi, M.B. Tahoori, "Soft Error Mitigation for SRAM-Based FPGAs," *23th IEEE VLSI Test Symposium*, pages 207-212, May 2005.

[7] E.S.S. Reddy, V. Chandrasekhar, M. Sashikanth, V. Kamakoti , "Online Detection and Diagnosis of Multiple Configuration Upsets in LUTs of SRAM-based FPGAs," *19th IEEE International Parallel and Distributed Processing Symposium*, pages 172-175, 2005.

[8] E.S.S. Reddy, V. Chandrasekhar, M. Sashikanth, V. Kamakoti , "Novel CLB Architecture to Detect and Correct SEU in LUTs of SRAM-based FPGAs," *IEEE International Conference on Field-Programmable Technology*, pages 121-128, December 2004.

[9] S.Srinivasan, A.Gaysen, N.Vijaykrishnan, M.Kandemir, Y.Xie, M.J. Irwin, "Improving Soft-error Tolerance of FPGA Configuration Bits," *IEEE/ACM International Conference on Computer Aided Design*, pages 107-110, Nov. 2004.

[10] Michael Wirthlin, Eric Johnson, and Nathan Rollins, "The Reliability of FPGA Circuit Designs in the Presence of Radiation Induced Configuration Upsets," *11th International IEEE Symposium on Field-Programmable Custom Computing Machines,* pages 113-122, 2003.

[11] F. Lima, L. Carro, R. Reis, "Designing Fault Tolerant Systems into SRAM-based FPGAs," *IEEE/ACM Design Automation Conference*, pages 650-656, June, 2003.