ISSN: 2395-1680 (ONLINE)
DOI: 10.21917/ijme.2016.0026

A CONCURRENT ERROR DETECTION SCHEME FOR TOTALLY SELF CHECKING FPGA LOOK-UP TABLE

S. Natarajan[1] and V. Deepa[2]
Department of Electronics and Communication Engineering, United Institute of Technology, India
E-mail: [1]deanece@uit.ac.in, [2]mail2deepa07@gmail.com

Abstract

Field Programmable Gate Arrays are widely useful in mission critical applications. FPGAs have fixed architecture; it has the capability to change function in situ for a particular application. SRAM based FPGAs are vulnerable to Single Event Upsets (SEUs), which poses unintended change to the logic functions on exposure. The project proposed is a unidirectional error detection scheme i.e., Scalable Error Detection Coding (SEDC) scheme, for use in FPGA Look-up tables. The SEDC check bits are generated along with the programming bits and it is stored on the FPGA SRAM cells during the normal operation of the LUTs. The programming bits are processed to check bit generator where corresponding code bits are generated for the programming bits. The newly generated code bits are compared with the pre-stored code bits. Any single or multiple unidirectional errors as a result of SEU is detected by this scheme. Scalability is the significant advantage of this scheme - it can be scaled to any input data length. With the increase in input data length, only the area gets scaled while the latency remains constant irrespective of the binary data length. The implemented algorithm achieves 100% error detection. The Proposed SEDC scheme is simulated using Tanner EDA tool and the layouts are generated using Microwind.

Keywords:

Look-Up Table (LUT)-Based Field Programmable Gate Arrays (FPGAs), SEDC Codes, Single Event Upsets

1. INTRODUCTION

In the last decade, advances in integrated circuit technology have inflated drastically the complexity of systems that can be realized as a ULSI scale single chip. This ongoing transition from traditional Application Specific Integrated Circuits to System-on-a-Chip has lead to new Challenges and paradigm shifts in design methods, system and circuit architectures and testing techniques. Advanced microelectronic technologies are becoming increasingly susceptible to faults and errors due to incident particles.

When high-energy neutrons and alpha particles strike a sensitive region in a semiconductor device, they generate a dense local track of electron-hole pairs. This may be collected by a p-n junction resulting in a current pulse of very short duration termed a single-event upset (SEU) in the signal value. A SEU may cause a bit flip in some latch or memory element thereby altering the state of the system resulting in an error. The minimum charge necessary to invert a logic state, or cause a fault, is called critical charge. If the fault is made visible to a user, it is then called an error. Not all faults manifest them as errors because of the functionality within the circuit.

The errors occurring in a system are Symmetric, Asymmetric, and Unidirectional. Symmetric errors are those if both 0 to 1 and 1 to 0 transitions occur simultaneously in a data word. Asymmetric errors are those if only 0 to 1 or 1 to 0 transitions are likely to occur, and the error type is known a priori. If both 0 to 1 and 1 to 0 transitions can occur in data word, but in any particular word if all errors are of same type, then the errors are called unidirectional errors. In Complementary Metal Oxide Semiconductor, faults can be of stuck-at-0 or stuck-at-1, which appear only as unidirectional errors [3]. In order to maintain an acceptable level of reliability, it is becoming mandatory to design ICs that are tolerant to these faults.

The rest of this paper is organized as follows: Section 2 reviews the previous Error Detection techniques. Section 3 introduces new SEDC scheme to Look Up Table. Section 4 and 5 describes the Generation and Formulation of proposed SEDC codes. Section 6 explains the System Implementation. Section 7 finally concludes the paper.

2. RELATED WORK

Error detecting codes are found to be more efficient that the other existing fault detection techniques in terms of area, speed and fault coverage and is most suited for implementing self-checking circuits [9][13]. Some of the unidirectional errors detecting codes are Parity code, Hamming code, Reed Solomon code, Berger code and Bose code [4]-[7]. Among the existing coding algorithms, Berger coding algorithm [7] alone has 100% fault coverage for all unidirectional errors. The check bit is computed either by counting the number of logic 1s in the data word and use the inverted binary count value as the code word or counting the number of logic 0s in the data word and use the binary count value as the code word. The drawback in Berger code is that the length of code word is increased, i.e., the check bit gets added with the information bits. Berger code word of length n bits having I information bits and k check bits, where $[k = \log_2(i + 1)]$, $n = i + k$.

3. SEDC BASED LOOK UP TABLE

The basic block of FPGA's based reconfigurable architecture is the Look-Up Table (LUT)-based function generator and it comprises of SRAM cells for storing values [2]. Different error detection methods are employed to detect errors in programming bits stored in SRAM cells. Among all fault detection technique, Error coding techniques are found to be more efficient.

The Fig.1 shows the architecture of Self Checking one LUT in FPGA. The new error detection algorithm is applied to the FPGA Look Up Table. The normal K-bit input LUT is indicated by a dashed box at the left and it requires $2^K = n$ programming bits. The SEDC check bits for F are pre-computed by the programming device and it is stored on FPGA during configuration, i.e., C = SEDC(F). The input 'F' is fed into SEDC check bit generator and it is encoded into SEDC code bits without affecting the FPGA

performance. Finally, the equality tester compares the online generated SEDC code bits with the pre-stored SEDC code bits. Any unidirectional error in '*F*' or '*C*' is detected and indicated by the equality checker by an error indication signal '*V*'. Same architecture can be used throughout the FPGA without changing the existing routing structure of the FPGA.

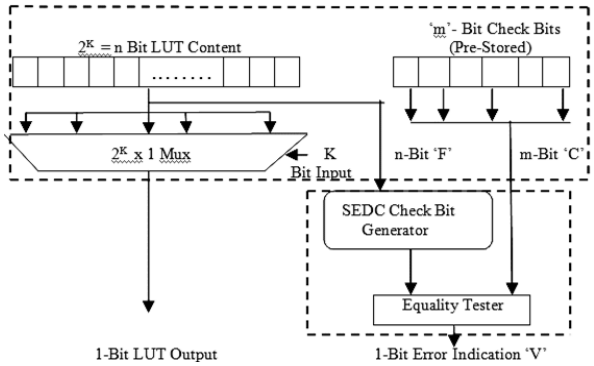

Fig.1. Architecture of Self Checking Look Up Table [2]

4. GENARATION OF PRE STORED SEDC CHECK BITS

Scalable Error Detection technique detects all unidirectional errors in the hardware. Scalable Error Detection Coding scheme is formulated and designed in such a way that only area is scaled, while latency depends on a small portion of the input data [1]. The algorithm partitions the given input data into 2-, 3- and 4-bit data and generates the corresponding SEDC code for the data.

Consider an input binary data '*F*'. The data length of '*F*' is *n*-bits, represented as (*F*0, *F*1, *F*2…..*Fn*-1). The two parameters '*a*' and '*b*' are generated using Eq.(1). The parameter '*a*' should always be a positive integer, and parameter '*b*' can have values of {2, 3, 4}.

$$a = \frac{n - \max(b)}{3} \tag{1}$$

Once the value for '*a*' is found and if it satisfies the condition of being an integer, the value for '*b*' is found. After calculating the values of '*a*' and '*b*', the n-bit binary input data is partitioned into '*a*' times of 3-bit segment and one '*b*'-bit segment, the partition is shown in Fig.2. The SEDC code bits are generated by using Eq.(2).

$$M = \lceil \log_2(n+1-3a) + 2a \rceil \tag{2}$$

Any integer value of '*n*' can be partitioned by this scheme. The Fig.2 represents the partition of input binary word of *n* bit length. The Table.1 represents the code bits for data length of *n* = 3 bits.

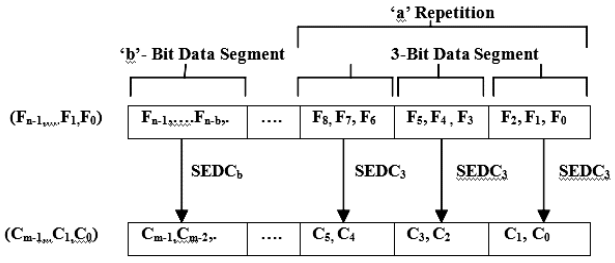

Fig.2. Partitioning of Input Data Word using SEDC scheme

Table.1. $SEDC_3$ Code Table

3 Bit Data	$SEDC_3$ Code Scheme1	$SEDC_3$ Code Scheme2
000	11	11
001	01	10
010	01	10
011	10	01
100	01	10
101	10	01
110	10	01
111	00	00

5. FORMULATION OF SEDC CODES

5.1 $SEDC_2$ CODING SCHEME

The Fig.3 (a) describes the 2-D illustration of $SEDC_2$ scheme, the data words are represented by nodes and their code words are written in brackets.

$$[C1{:}C0] = [\text{NAND}(F1, F0) : \text{XNOR}(F1, F0)] \tag{3}$$

By using Eq.(3) different code words are assigned to binary data word. The input data words are represented by *F*0, *F*1, while the code bits are symbolized by *C*0, *C*1.

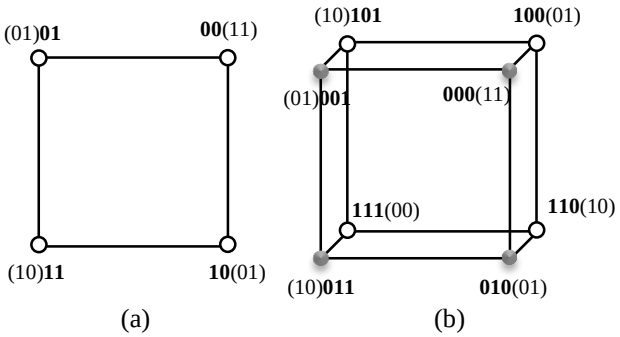

Fig.3. (a) 2-D illustration of $SEDC_2$ (b) 3-D illustration of $SEDC_3$ [2]

5.2 $SEDC_3$ CODING SCHEME

$SEDC_3$ code [2] for 3-bit data is computed as per Eq.(4).

$$(C1, C0) = \begin{matrix} SEDC_2(F1, F0) & \text{if } F2 = 0 \\ SEDC_2(\bar{F}1, \bar{F}0) & \text{if } F2 = 1 \end{matrix} \tag{4}$$

The Fig.3(b) describes the 3-D illustration of $SEDC_3$. Same notations are used in Fig.3(b) as in Fig.3(a). whenever, there is a unidirectional change of 2-bits in data word '001' to '111' (two MSB's changing from '00' to '11'), the code word reflect the change in the opposite direction (LSB of the code changes from '1' to '0').The first four code words of $SEDC_3$ are same as $SEDC_2$ as described in Fig.3(a). The other code words are formulated as follows,

1. The given data word must be 1's complemented (assume '101'→'010').

2. The $SEDC_2$ code word is found to the resultant complemented data (‘010’→‘10’).
3. The obtained $SEDC_2$ code word is 1’s complemented (‘10’→‘01’).
4. The result code that is obtained in step 2 becomes the code word for the actual data word (‘101’ [$SEDC_2$] ‘10’).

5.3 $SEDC_4$ CODING SCHEME

$SEDC_4$ code [2] for 4-bit data word is formulated using Eq.(5).

$$[C2:(C1, C0)] = [\text{NOT}(F3):SEDC_3(F2, F1, F0)] \quad (5)$$

5.4 $SEDC_n$ CODING SCHEME

It is found that each data word and corresponding code bit is independent of each other. This independency makes the proposed SEDC scheme scalable. Here initially the *n*-bit binary input data is partitioned into ‘*a*’ times of 3-bit segment and one ‘*b*’-bit segment, and then these segments are encoded using SEDC ‘*b*’ and ‘*a*’ times of $SEDC_3$ code generators in parallel (Fig.2). Scalability is a unique feature of this scheme.

6. SYSTEM IMPLEMENTATION

The schematic of proposed LUT structure are designed and implemented by using TANNER EDA tool. Circuit verification is done on the TANNER EDA tool. Schematic of the Self Checking LUT are designed on the S-Edit and Netlist simulation done by using T-spice and waveforms are analyzed through the W-edit. The Technology File that is included in Tanner is 130nm.In this project we used DSCH [Digital Schematic] software for simulating the circuit and to generate the code, MICROWIND is used to extract the layout of the schematic diagram. The simulation is performed for LUT content of 2 bits to 32 bits and their corresponding layouts are also extracted. The result proves that the proposed SEDC algorithm is more efficient in terms of detection than the other existing error detection schemes. The Schematic and layout for Binary word length of 2 bits is as follows,

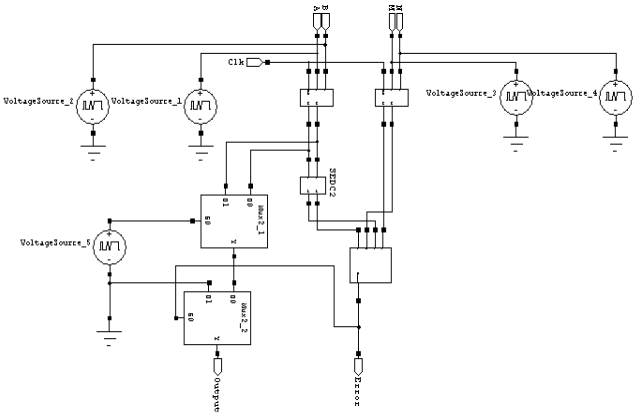

Fig.4. Schematic of LUT Content = 2bits in Tanner

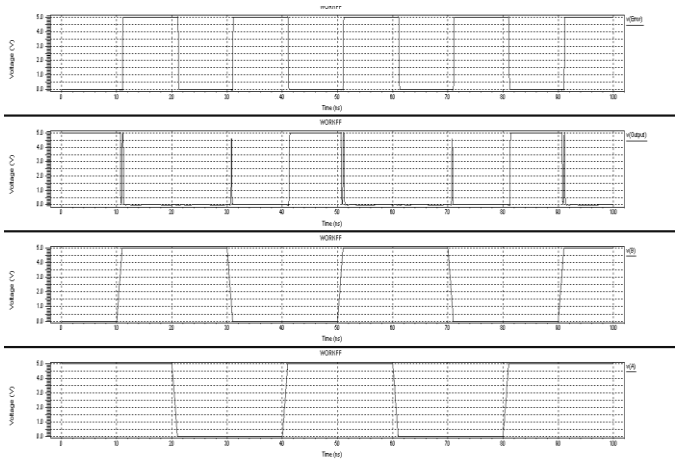

Fig.5. Waveform of LUT Content = 2bits

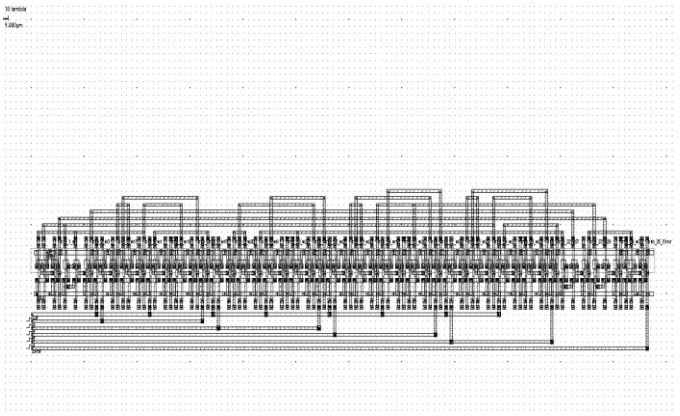

Fig.6. Layout of LUT Content = 2bits in Microwind

7. CONCLUSION

The New Concurrent Error Detection Scheme-Scalable Error Detection Coding (SEDC) is formulated and applied to Look Up Table. By the proposed new technique it is shown that the LUT are Self-Checking and Fault Tolerant. The proposed SEDC algorithm is found to be significantly efficient than the existing unidirectional error detection scheme in terms of detection. The SEDC architecture is implemented and simulated using Tanner EDA tool and the layouts are generated by means of Microwind. The proposed scheme is Scalable to any input binary data length and it achieves 100% error detection.

REFERENCES

[1] Natarajan Somasundaram, Jeong A Lee, Farhad Mehdipour, Ramadass Narayanadass and Y.V. Ramana Rao, “Scalable Error Detection Coding Algorithm for Totally Self-Checking (TSC) Circuits”, *Consumer Electronics Times*, Vol. 2, No. 3, pp. 116-123, 2013.

[2] Jeong-A Lee, Zahid Ali Siddiqui, Natarajan Somasundaram, and Jeong-Gun Lee, “Self Checking Look-Up Table Using Scalable Error Detection Coding Scheme”, *Journal of Semiconductor Technology and Science*, Vol. 13, No. 5, pp. 415-422, 2013.

[3] J.R. Schwank, *et al.*, "Radiation Effects in MOS Oxides", *IEEE Transactions on Nuclear Science*, Vol. 55, No. 4, pp. 1833-1853, 2008.

[4] P. Kubalík and H. Kubátová, "Parity Codes Used for On-Line Testing in FPGA", *Acta Polytechnica*, Vol. 45, No. 6, pp. 53-59, 2005.

[5] Steven W. Bums and K. Niraj, "A Totally Self-Checking Checker for a Parallel Unordered coding scheme", *IEEE Transactions On Computers*, Vol. 43, No. 4, pp. 490-495, 1994.

[6] X. Kavousianos, D. Nikolos and G. Sidiropoulos, "Novel Single and Double Output TSC CMOS Checkers for m-out-of-n Codes", *VLSI Design*, Vol. 11, No. 1, pp. 35-45, 2000.

[7] D.A. Pierce Jr. and P.K. Lala, "Modular Implementation of Efficient Self-Checking Checkers for the Berger Code", *Journal of Electronic Testing: Theory and Application*, Vol. 9, No. 3, pp. 279-294, 1996.

[8] Fernanda de Lima, Luigi Carro and Ricardo Reis, "Single Event Upset Mitigation Techniques for SRAM-Based FPGAs", Available at: http://www.inf.ufrgs.br/~fglima/lima_latw03.pdf

[9] Niraj K. Jha, and Sying-Jyan Wang, "Design and Synthesis of Self-Checking VLSI Circuits", *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, Vol. 12, No. 6, pp. 878-887, 1993.

[10] Tatjana R. Stankovic, Mile K. Stojcev and Goran L. Djordjevic, "Design of Self-checking Combinational Circuits", *Proceedings of 6th International Conference on. Telecommunications in Modern Satellite, Cable and Broadcasting Service*, Vol. 2, pp. 763-768, 2003.

[11] S. Mitra and E. J. McCluskey, "Which Concurrent Error Detection Scheme To Choose?", *Proceedings of International Test Conference*, pp. 985-994, 2000.

[12] P. Lala, "*Self-Checking and Fault-Tolerant Digital System Design*", Francisco: Morgan Kaufman Publishers, 2001.

[13] C. Bolchini, F. Salice and D. Sciuto, "Designing Self-Checking FPGAs through Error Detection Codes", *Proceedings of 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 60-68, 2002.