International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 10 | Oct -2017 www.irjet.net p-ISSN: 2395-0072

Counter Matrix Code for SRAM Based FPGA to Correct Multi Bit Upset Error

Elakkiya B[1], Chandran V[2]

[1]*PG Student, Dept of ECE, Government College of Technology, Tamilnadu, India*
[2]*Assistant Professor, Dept of ECE, Bannari Amman Institute of Technology, Tamilnadu, India*

Abstract - *Memory blocks are the most significant features of any design, frothy of its silicon area, functionality and dependency. SRAM memories are the main benefactors to the Soft Error Rate of the system. Since error detecting and correcting codes are commonly available and especially effective against most types of Single Event Effects, Multiple Bit Upsets and advanced errors gathering may conquer the error correction capabilities. Existing techniques use error correction codes with significantly high overhead to reduce MBUs in configuration frames, here a low-cost error-detection code to detect MBUs in configuration frames as well as a Counter Matrix Code (CMC) based on correction method is proposed. The proposed plan does not demand any alteration to the FPGA architecture. Implementation of the proposed plan on a Xilinx Virtex-6 FPGA devices displays that it can detect 100% of MBUs in the configuration frames.*

Key Words: **Multiple Bit Upsets (MBU), Conquer, Counter Matrix Code (CMC), FPGA.**

1. INTRODUCTION

SRAM based FPGAs are largely used in a variation of application domains because of its minimum time-to-market time, pliability, high density, and cost efficiency. Nevertheless, increasing transistor count per chip (i.e., Moore's law) copulate with the decreased operating voltage in the past years results in an exponent growth in soft error rate (SER) of digital circuits. In order to encounter the ever-increasing performance and power demands, FPGAs are typically constructed using the majority enhanced technology nodes. Newly, FPGAs based on a 14-nm technology with denser integration schemes, such as 3-D die stacking, have been proclaimed. In such small device geometries, a single radiation-induced particle strikes likely to incline several adjacent cells in a memory array, leading to a multiple bit upset (MBU).Considering the fact that the MBU rate in Nano. Scales is comparable with the single event upset (SEU), an appropriate plan is necessary to detect and correct multiple errors in memory arrays. Especially SRAM-based FPGAs are more prone to soft errors as a particle strikes in a configuration frame.

Several plans have been broadcasted to address the increasing soft error in the FPGA configuration frames. The main aim of these plans is to reduce error latency, and hence, to avoid error gathering within configuration frames However, accumulated errors in both data and configuration bits histrionically limit the mean time to failure of such schemes. Another technique is to optimize the configuration frame circuitry for soft errors. However, such hardening techniques are not implemented in the existing FPGA devices because of their excessive area overheads. Therefore, a low-cost solution is required to correct erroneous configuration frames during operation. The Fusion of configuration scrubbing and error correction codes (ECCs) is a powerful solution to detect and correct radiation-induced transient errors in configuration bits.

There are also a few schemes to specifically address MBUs in FPGA devices. The scheme proposed in 2-D Hamming code in each configuration frame to correct MBUs. Another scheme based on interleaved single error correction Hamming code has been presented that could correct up to four adjacent error bits. Based on the fact that error detection can be done at much lower cost than error correction we propose the MBU detection technique to detect the erroneous configuration frame. For detecting MBUs in the configuration frames of the FPGA, we propose a low cost technique, namely, interleaved n dimensional (InD) parity. The interleaving distance of this technique is to make more efficient based on the actual MBU patterns and their respective probabilities obtained from a detailed technology-dependent theory. Furthermore, by carefully dividing the frames into several clusters, the proposed technique can detect and correct an MBU affecting several adjacent configuration frames.

2. BACKGROUND

2.1 Multiple Bit Upsets (MBUs)

Multiple Bit Upsets (MBUs) are a major reliable concern in Nano scale technology nodes. Appearing such errors in the configuration frames of a field programmable gate array (FPGA) device forever affects the functionality of the mapped design. Periodic configuration scrubbing Fused with a low-cost error correction scheme is an efficient approach to avoid such a permanent effect. Existing methods use error correction codes with considerably high overhead to reduce Multiple Bit Upsets in configuration frames. For error detecting Counter Matrix Code (CMC) is proposed and calculating the parity bits for horizontal and vertical bits. More over the detecting errors are corrected by simple syndrome comparison method. To reduce the error recovery

time and increase the detection coverage, frames are divided into several clusters by CMC techniques.

In order to quantify the MBU correction capability, a 3-D-TCAD based neutron particle strike simulation is conducted by using a commercial soft error assessment tool. The SPICE net list and the memory layout and also the radiation environment information are prepared as inputs to the tool to calculate the distribution of produced current pulses for each cell according to a nuclear database. After hand, the SEU and MBU rates are infused by injecting the gained current pulses in the SPICE net list. Using this commercial tool, we have acquired the occurrence probabilities of neutron-induced MBU patterns in the terrestrial environment on an SRAM memory designed for a45-nm technology. Multiple Bit Upset size distribution and its patterns with high occurrence probabilities, almost half of the SRAM soft errors in the employed 45-nm technology are MBU events. In this theory, the largest MBU size observed affects 24 bits. For smaller technology nodes, this ratio and also the size of largest MBU further grow. This clearly shows the importance of protecting configuration frames against MBU events.

2.1 FPGA Technologies

FPGA (Field Programmable Gate Array) is a coordinated circuit having gate matrix which can be programmed by the consumer in the field lacking costly equipments. An FPGA appointing a set of programmable logic gates and abounding interconnect resources, which is feasible to implement complex digital circuits. FPGAs have an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that employ the blocks to be "wired together", like many logic gates that can be inter-wired in different arrangements. Logic blocks can be configured to perform complex combinational functions, or entirely uncomplicated logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. The FPGA configuration is commonly specified using a hardware description language (HDL),alike to that performed for an application-specific integrated circuit (ASIC). FPGA Implementation Technologies Configuration bit stream can be stored in FPGA using different technologies. The majority of FPGAs is based on SRAM (Static RAM) memory.

2.1.1 SRAM-Based FPGAS

The configuration memory of FPGAs is organized into configuration frames that are the smallest addressable units and compose the majority of SRAM cells in FPGAs. The size and the number of the configuration frames vary from one device to another. SRAM-based FPGA keeps logic cells configuration data in the static memory (organized as an array of latches). Since SRAM is violent and can't keep data without power source, such FPGAs must be programmed (configured) upon start. There are two basic modes of programming:

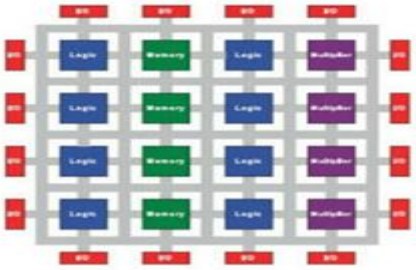


Fig -1: SRAM based FPGA Overview

I) Master Mode

FPGA reads configuration data from an external source, so as an external Flash memory chip.

II) Slave Mode

FPGA is arranged by an external master device, such as a processor. This can be usually done by the way of a dedicated configuration interface or via a boundary-scan (JTAG) interface. SRAM-based FPGAs include majority chips of Xilinx Virtex, Spartan families, Altera Stratix and Cyclone. RAM-based FPGAs with an internal flash memory character of FPGA is commonly as the previous, exclude that these chips has an internal flash memory blocks, thus removing the need to have an external non-violent memory. The memory cell of SRAM constitutes of Vcc, Vdd, bitline, ~bitline, word line one such example of FPGAs is the Xilinx Spartan-3AN family. Each design of Spartan-3AN has an in chip flash memory module with an SPI interface is efficient of storing two or more configuration bit streams. The bit stream can be chosen during startup. Another example of such technology is the Lattice XP family by Lattice Semiconductors. Using internal non-violent memory can be also useful to prevent no authority bit stream copying.

2.1.2 Flash-Based FPGAs

The flash-based FPGAs shouldn't be muddled with the previous type. The SRAM-based FPGAs with internal flash memory use flash only at startup to load data to the SRAM configuration cells. On the adverse, true flash-based FPGA uses flash as a earliest formed resource for configuration storage, and doesn't need SRAM (a identical technology is employed in CPLDs – complex programmable logic devices, however the FPGA architecture is truly distinct from that of CPLD). This technology has a Favorable of being less power consumptive. Flash-based FPGAs are also more tending to permit radiation effects. As in the previous case, using flash-based FPGAs can be a solution to stop unauthorized bit stream copying.

2.1.3 Antifuse-Based FPGAs

Antifuse-based FPGAs are not the same from the previous ones in that they can be programmed only one time. The Antifuse is a device that doesn't act current initially, but can be burned to conduct current (the Antifuse behavior is thus opposite to that of the fuse, since the name). The Antifuse-based FPGA can't be then reprogrammed because there is no way to put back a burned Antifuse into the initial state. Antifuse-based device families constitutes of Accelerator produced by Actel.

3. ERRORS

Many communication channels are matter to channel noise, and so errors may be introduced during transmission from source to destination. Errors are of two types such as soft error and firm error. A soft error is said to be a "glitch" in a semiconductor device. These glitches are random, usually not catastrophe, and normally do not damage the device. They are caused by external elements outer surface of the designer's control. Many systems can allow some level of soft errors. Errors introduced because of the external radiation or electrical noise rather than the model or manufacturing defects are known as soft errors.

An error in a memory element is declared soft since it spoils the data. This identical type of radiation induced error in an FPGA is a "firm" error, since it is not just a transient data error. When a firm error appears, the data is not corrupted; it is the device's configuration or "personality" that is affected. The error changes the actual function of the device. Soft errors in SRAM memories can be either Single-Event Upset (SEU), where an ionizing particle damages a single bit or Multiple-Bit Upset (MBU), where more than one bit is upset on a single measurement. Relying on the underlying technology and the incident particle, different types of multiple-bit errors are possible. It has been shown that incident neutron particles can react with the die contaminant and generate secondary particles with sufficient energy to make multiple errors.

4. RECOVERY BASED ON CMC

In proposed CMC (Counter Matrix Code) Vertical and horizontal parities are calculated but the number of parity bits are less. Parity bit is nothing but the redundant bits used to recover or extract the original bit from the soft error affected bits i.e. error bit. For 32 bit data, total numbers of parity bits are 24. In original data and the error data which is retrieved from memory (error data) parity bits are being calculated in order to correct the errors. The block diagram of counter matrix code is shown in Fig. 2. D_{in} represents the original data, which is to be stored in SRAM memory. D_{err} indicates the bits, assumed which is affected by soft error, when it is stored in memory.

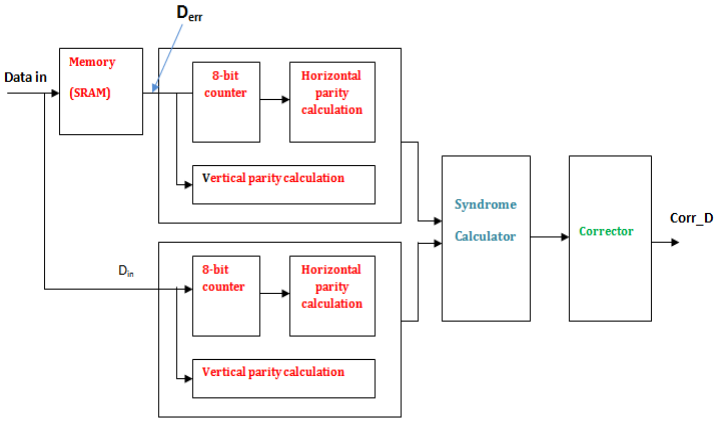


Fig -2: Counter Matrix Code bock diagram

In CMC encoder block, one 8-bit counter is used to count the number of ones in corresponding segmented positions. Our decoder block consists of both syndrome calculator and predictor. Syndrome calculator is the error finder which is probably used to correct and check the bits. So in memory the error data is said to be corrected using counter matrix code method, thus receiving the corrected data. The original data and the corrected data should be configured with the same parity sense, hence the correction coverage is said to be maximum.

5. SIMULATION RESULTS

In this section, the various parameters and the simulation diagrams of proposed system is discussed. Meanwhile, the parameter of proposed method is compared with our existing method.

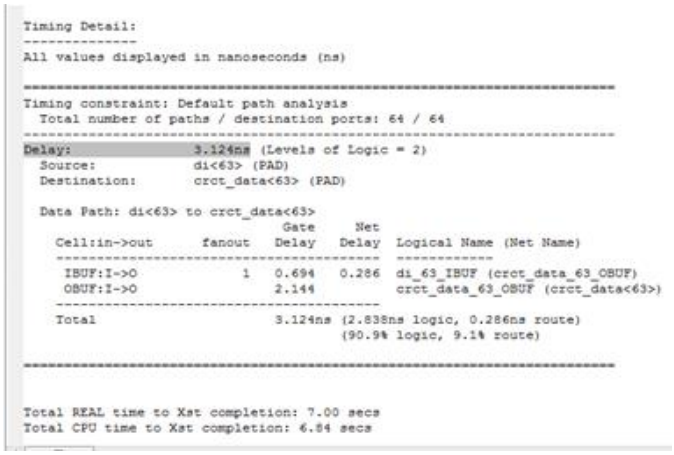
Table -1: comparison between existing method and proposed method

Parameters	Existing (DMC)	Proposed (CMC)
Number of parity bits(for 64 bits)	72 bits	40 bits
Delay	0.665 ns	0.345 ns
Power	2.586 w	1.293 w

Table -2: comparison between existing method and proposed method

Parameters	Existing method	Proposed method
Number of parity bits(for 32 bits)	36 bits	24bits
Delay	1.33 ns	0.345 ns
Correction coverage	95%	100%

Table.1 and Table.2 shows the various parameter comparisons between existing and proposed method. From Table.2, the cost of our CMC (proposed) method is more much better compare than the existing method. Number of parity bits required for existing and proposed method is 36 bits and 24 bits respectively. From the above statement the number of bits saving is 12 bits for 32 bit original data. In the simulation discussion in terms of delay and detection, the input is taken as 64 bits. Figure.3 shows the timing report for our proposed method. Delay generated by our proposed method is 3.125ns. The detection of errors or correction coverage of error data is 100%.



```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 64 / 64
-------------------------------------------------------------------------
Delay:               3.124ns (Levels of Logic = 2)
  Source:            di<63> (PAD)
  Destination:       crct_data<63> (PAD)

  Data Path: di<63> to crct_data<63>
                             Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             1   0.694   0.286  di_63_IBUF (crct_data_63_OBUF)
     OBUF:I->O                 2.144          crct_data_63_OBUF (crct_data<63>)
    ----------------------------------------
    Total                      3.124ns (2.838ns logic, 0.286ns route)
                                       (90.9% logic, 9.1% route)

=========================================================================


Total REAL time to Xst completion: 7.00 secs
Total CPU time to Xst completion: 6.84 secs
```

Fig -3: Timing report of CMC method

Fig.3. shows the timing report of CMC method. The highlighted portion of fig.3.indicates the total time taken by our proposed method to recover the error from errorous data.

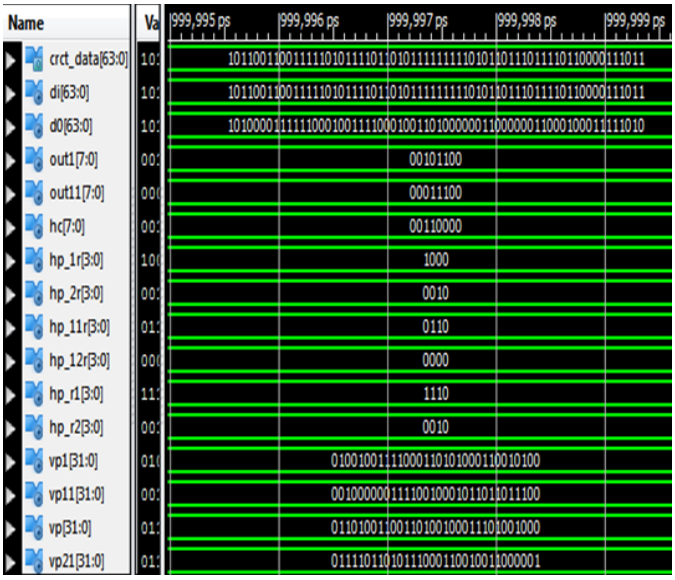


Fig -4: Output of Counter Matrix Code

Fig.4.shows the simulation output of our proposed counter matrix code. It is clearly provided that the error correction coverage is better when compared to the existing techniques for multi bit upset corrections and it also corrects single bit upset and multi bit upset.

6. CONCLUSION

Radiation-induced Multiple Bit Upsets are a serious reliability concern in Nano scale technology nodes. The configuration frames are the most vulnerable resources on the FPGA fabric to soft errors as they constitute the majority of the FPGA memory bits and once affected by soft errors, permanently change the functionality of the mapped design. The obtained results showed that the proposed technique can detect and correct single bit upsets as well as the Multiple Bit Upsets. (MBUs).Error detection coverage is high in our proposed technique compared to existing technique.

REFERENCES

[1] Parthasarathy M.B. Rao Mojtaba Ebrahimi Razi Seyyedi Mehdi B. Tahoori, " Protecting SRAM-based FPGAs Against Multiple Bit Upsets Using Erasure Codes," IEEE'2014.

[2] Jing Guo, Liyi Xiao, Zhigang Mao, Member, and Qiang Zhao, " Enhanced Memory Reliability Against Multiple Cell Upsets Using Decimal Matrix Code, " IEEE'2014.

[3] Mojtaba Ebrahimi, Seyed Ghassem Miremadi,Hossein Asadi, Member, and Mahdi Fazeli, " Low-Cost Scan-Chain-Based Technique to Recover Multiple Errors in TMR Systems, " IEEE'2011.

[4] Costas Argyrides, Dhiraj K. Pradhan, and Taskin Kocak, " Matrix Codes for Reliable and Cost Efficient Memory Chips, " IEEE'2011.

[5] Eishi Ibe,Hitoshi Taniguchi,Yasuo Yahagi,Ken-ichi Shimbo,and Tadanobu Toba, " Impact of scaling on Neutron-Induced Soft Error in SRAMs From a 250nm to a 22nm Design Rule, " IEEE'2010.

[6] Jeff Draper and Riaz Naseer, " Parallel Double Error Correcting Code Design to Mitigate Multi-Bit Upsets in SRAMs, " IEEE'2008.