A Review of the Most Adopted Fault Tolerance Approaches for SRAM based FPGA Real Time Critical Embedded System

Meryem Bouras, Hassan Berbia
ENSIAS, Mohammed V University in Rabat, Agdal Rabat BP 713, Rabat, Morocco

Keywords: ADCS, Fault tolerance, Reliability, SRAM based FPGA, Configuration memory scrubbing

Abstract: Since the downscaling in technology FPGAs became widely used in space embedded systems, due to their high computing capability, flexibility and easiness of implementation. Especially, SRAM-based FPGAs are now more adopted and implemented; as processing unit; in digital systems like the ADCS, in nanosatellite. However, this continuous downscaling made radiation-induced errors become a major concern. Particularly for SRAMs-based FPGAs, because they are more sensitive to radiations and more prone to soft errors. Regarding this matter, many approaches were developed and adopted in literature. Our objective is to optimize the reliability of the ADCS system, by improving the reliability of its SRAM based-FPGA. Therefore, in this paper we briefly describe the SRAM based FPGA configuration memory. Also, we present the different faults tolerance approaches proposed in the literature, and we evaluate these approaches by illuminating their advantages and disadvantages.

1 INTRODUCTION

Attitude estimation and control of a rigid body, in domain like Aerospace, Aeronautics and Robotics, is an important subject discussed by the scientific comity in the last decades. Those domains, especially Aerospace, requires designing reliable systems, that provides an accurate estimation of the attitude, space-saving and autonomy, and therefore less energy consumption.

Environmental factors significant to space radiation hazards are: trapped electrons and protons of the Van Allen radiation belts, transient solar energetic particles and galactic cosmic rays. These hazards affect or profoundly damage spacecraft electronics and microelectronic parts and cause a limited number of dominant radiation effects. The two most discussed effects are the Total Ionizing Dose (TID) it could be contained by shielding, and the Single event effect (SEE), it is an anomaly caused by a single energetic particle striking a device.

The SEE phenomenon can be classified into various types: Single Event Upset (SEU), it is undesirable change in the logic state of the device. Single Event Transient (SET), it is one or more voltages pulses (i.e. glitches) that propagate through the device, if it results in an incorrect value being latched in a sequential logic unit, then it is considered as an SEU. These soft errors can cause flipping of single bit or more often adjacent bits in the memory. Single Event Latchup (SEL), it may lead to permanent damage to the device, and the most dangerous type is Single Event Burnout (SEB), that leads to permanent failure. These soft errors affect the Attitude determination and control system (ADCS) rigorous spacecraft attitude control and manoeuvres, influences its reliability and compromise the satisfactory and the safety of a nanosatellite mission. As mentioned in the paper (Bouras et al., 2016), our main objective is to optimize the reliability of this system, by improving the reliability of its processing unit. Static Random- Access Memory (SRAM) based Field-Programmable Gate Arrays (FPGAs) are more adopted and implemented; as processing unit; in digital systems like ADCS, because of their low cost, ability of reconfiguration and higher performance. However, the continuous downscaling made the SRAMs-based FPGAs, more sensitive to soft errors like SEU, because the configuration memory that defines the circuitry implemented inside them, is more prone and vulnerable to Single Bit Upset (SBU) and Multiple Bit Upset (MBU) (Colodro-Conde and Toledo-Moreo, 2015). Therefore, to protect this memory, and to minimize these two faults, SBU and MBU, many approaches were developed and adopted

Bouras, M. and Berbia, H.
A Review of the Most Adopted Fault Tolerance Approaches forSRAM based FPGA Real Time Critical Embedded System.
DOI: 10.5220/0009775701020106
In *Proceedings of the 1st International Conference of Computer Science and Renewable Energies (ICCSRE 2018)*, pages 102-106
ISBN: 978-989-758-431-2

in the literature. such as, optimized ECC (error correcting code) (Dutta and Touba, 2007), Interleaving (Xavier and Kantham, 2013), and Hamming EDAC (error detection and correction) cores with SEC-DED (single-error-correction, double-error-detection) (Lankesh and Narasimhamurthy, 2015), Scrubbing (readback scrubbing, ECC scrubbing, CRC scrubbing) (Siegle and Vladimirova, 2015). These techniques capabilities have proven to be an effective way to protect the configuration memory (Colodro-Conde and Toledo-Moreo, 2015).

In this article we will we briefly describe the SRAM based FPGA configuration memory. Also, we will present, define and review state of the art faults tolerance methods used for detecting and correcting SBUs and MBUs, and we will give a simple definition of the approach, that we chose among them. Then we will present a comparative study of these methods by illuminating their advantages and disadvantages.

2 SRAM BASED FPGA CONFIGURATION MEMORY

The SRAM based FPGA configuration memory can be affected by radiation effects. However, its ability to be updated in later design and mission stages, or to be reconfigured in-flight. Makes it more adopted in research and space applications. The most susceptible part of the configuration memory is the Bitstream. The Bitstream is a set of reprogrammable and volatile configuration memory bits, responsible for the configuration of all the SRAM based FPGA components, such as CLBs, routing, Block RAMs, DSP blocks and IO blocks. The size of the bitstream depends on the FPGA device and the considered application (Xilinx, 2005). The most popular SRAM based FPGA are Virtex FPGAs series by Xilinx, and the most adopted in literature is the Virtex 5.

A Virtex-5 FPGA configuration memory or bitstream is composed of 41 words of 32 bits (1,312 bits). Basically, the FPGA configuration memory floorplan is organized in rows and columns. Each column defines a specific type of resource and the rows divide each column in equal groups of elements. For example, one row of a CLB column is composed of 20 CLBs for the Virtex-5 FPGA (Tonfat et al., 2015). A graphical description of the organization of the floorplan is shown in Fig.1(fig 4 in (Tonfat et al., 2015)).

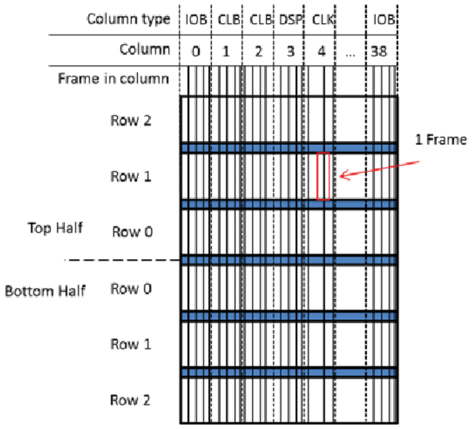

Figure 1: An example of the configuration memory Floorplan (fig 4 in (Tonfat et al., 2015)).

3 FAULT TOLERANCE TECHNIQUES

3.1 Related Work

With the continues downscaling in technology, memory density increases, soft error rate has drawn a major attention as the numbers of fault in the devices have increased significantly. Therefore, the tolerance of SEUs and MBUs effects on SRAM based FPGA has been an active subject of research, and many approaches for SEUs mitigation have been developed. These approaches can be categorized into SEC-DED (single error correcting-double error detecting) codes techniques ((Dutta and Touba, 2007), (Xavier and Kantham, 2013), (Lankesh and Narasimhamurthy, 2015)) and scrubbing techniques (Colodro-Conde and Toledo-Moreo, 2015), (Siegle and Vladimirova, 2015), (Legat et al., 2012), (Wirthlin and Harding, 2016), (Herera-Alzu and López-Vallejo, 2013)).

Conventional ECC techniques used in memories cannot correct MBUs caused by SEU (Dutta and Touba, 2007). Therefore, they proposed in (Dutta and Touba, 2007), a methodology based on deriving an error correcting code (ECC) through heuristic search technique, that can detect and correct the most likely double bit upsets while minimalizing the mis-correction probability of the improbable double bit upsets. The proposed ECC can be used in addition to bit interleaving or instead of it, to provide better protection from MBUs. However, it costs little more than the single error correcting-double error detecting (SEC-DED) codes commonly used.

Interleaving technique has been used to restrain MBUs. This technique rearranges cells in the physical arrangement to separate the bits in the same logical word into different physical words (Satyanarayana et al., 2014). However, interleaving technique may not be practical because, it is complex and requires higher memory.

Since Interleaving and built-in current sensors (BICS) have been successful in the case of single event upset (SEU) (Xavier and Kantham, 2013), in (Xavier and Kantham, 2013) they present an alternative approach to protect memories by using built-in current sensors (BICS) that can deduct errors by detecting changes in the current. They optimized the protection by proposing specific error correction codes (ECC) to protect memories against multiple-bit upsets. The method was evaluated using fault injection experiments.

Hamming codes are widely used for the single bit error correction double bit error detection (SEC-DED) which occurred during data transmission process. However, they cannot correct MBUs caused by SEU. In paper (Lankesh and Narasimhamurthy, 2015), they present an enhanced technique to detect double adjacent bit errors and to correct all possible single bit errors in Hamming codes through selective bit placement technique for memory application. This technique improves the probability of detecting double adjacent bit errors and provides a simple method of detecting double adjacent bit errors as compared to convolution coding through interleaving.

The most adopted approach is configuration scrubbing, this well-known memory scrubbing technique is adopted to mitigate upsets (SBU and MBU) in the configuration memory of SRAM-based FPGAs, by rewriting the configuration data, without interrupting the normal FPGA operation. The circuit that performs scrubbing is commonly named scrubber, there are two types of scrubber, internal and external scrubber. The scrubbing process of each one of them, can be implemented in software with high flexibility but with lower energy efficiency and lower configuration speed or hardware with high configuration speed and high energy efficiency. Both these scrubbers have been proven to be affective (Colodro-Conde and Toledo-Moreo, 2015), (Siegle and Vladimirova, 2015), (Brosser et al., 2014), (Legat et al., 2012), (Wirthlin and Harding, 2016) (Herera-Alzu and López-Vallejo, 2013). However, each one of them has his cons and pros, in (Berg et al., 2008) they give a detailed comparison. The external scrubber is implemented in external FPGA (Wirthlin and Harding, 2016). In the opposite, the internal scrubber, is more effective, with high efficiency, in matter of time and area because it's implemented inside the configuration memory of the SRAM based FPGA. Beside this classification by implementation methods, scrubbing can be classified by granularity (frame level oriented, device oriented, or module oriented), also it can be classified by correction mechanisms (blind scrubbing, readback scrubbing) (Tonfat et al., 2015).

3.2 Configuration Scrubbing Mechanisms

There are different scrubbing mechanisms (Siegle and Vladimirova, 2015), such as blind scrubbing, readback scrubbing. A detailed description is shown in Fig.2, and a comparison is giving in Table.2. A more detailed definition and comparison of these mechanisms can be found in (Siegle and Vladimirova, 2015).

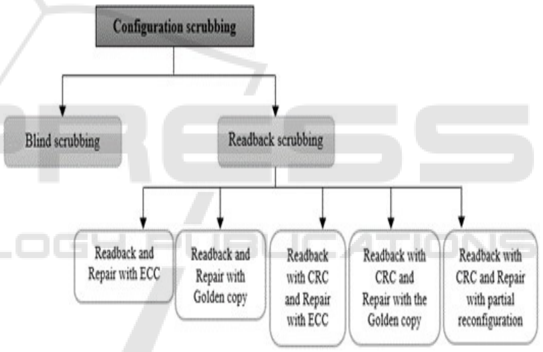
Configuration scrubbing
Blind scrubbing
Readback scrubbing
Readback and Repair with ECC
Readback and Repair with Golden copy
Readback with CRC and Repair with ECC
Readback with CRC and Repair with the Golden copy
Readback with CRC and Repair with partial reconfiguration

Figure 2: Different Scrubbing mechanisms adopted in literature.

3.2.1 Blind Scrubbing

Blind scrubbing or Open-loop scrubbing (Herera-Alzu and López-Vallejo, 2013) is a preventive mechanism performed without prior detection, to correct SBU and MBU cyclically through dynamic partial reconfiguration of the full configuration memory.

3.2.2 Readback Scrubbing

Readback scrubbing or Closed-loop scrubbing is a detection-correction technique, that combines both readback and scrubbing processes. After the configuration of the FPGA, the efficient data is loaded and stored as a golden copy in a PROM or flash ROM, then the readback is performed periodically, then if an upset in the memory is detected, an on-demand scrubbing is enabled (Siegle and Vladimirova, 2015), (Legat et al., 2012). These two processes can be achieved respectively using different methods such us, Readback and Repair with ECC which is adopted by Xilinx in most of its FPGAs (Xilinx, 2005), Readback and Repair with Golden copy (Herera-Alzu and López-Vallejo, 2013), Readback with CRC (cyclic redundancy check) and repair using ECC (Xilinx, 2010) or using the golden copy or using partial reconfiguration(Yang and Kwak, 2015).

- Readback and Repair with ECC, the ECC bits are used to locate the upsets during the readback, the repairing is also accomplished frame by frame using ECC. However, the process is rather complex and only single upsets can be corrected.
- Readback and Repair with Golden copy which is the simplest one, the concept relies on the comparison of readback data against the stored golden copy. In case there is a difference, the golden copy is used to scrub the faulty data in the configuration memory. This method needs the permanent availability of the golden copy of the configuration memory, similarly to the blind scrubbing with the corresponding power consumption.
- Readback with CRC and Repair with ECC and the Golden copy or partial reconfiguration is another possibility, which is to check the CRC computed value of the efficient data read from the configuration memory during the readback process. In case an upset is detected. Once located, the faulty data is scrubbed with the corrected data using ECC, the process is complex. The scrubbing can also be performed using the golden copy, however it requires more area. The repairing can also be realised by partial reconfiguration, which need less time.

4 COMPARISONS OF FAULT TOLERANCE TECHNIQUES

In this section we provide a simple comparison between the previous cited faults tolerance approaches in table 1. Also, we evaluate the advantages and disadvantages of the different configuration scrubbing mechanisms adopted in the literature in table 2.

Table 1: Advantages and disadvantages of Faults tolerance methods.

Faults tolerance methods	Advantages	Disadvantages
Optimized ECC	-Detects and corrects SBU	-Expensive than conventional ECC
Interleaving	-Detects and corrects SBU	-complex -requires higher memory -cannot detect MBU
Interleaving and built-in current sensors (BICS)	-Detects and corrects SBU -Detects MBU	-Complex -requires higher memory -expensive
Optimized Hamming codes	-Detects and corrects SBU -Detects MBU	-Less complex than interleaving
External scrubbing	-Detects and corrects SBU -Detects MBU -high configuration speed -high energy efficiency	-Area overhead -Time overhead
Internal scrubbing	-Detects and corrects SBU -Detects MBU -high configuration speed -lower energy consumption	-Less area overhead -Less time overhead

Table 2: Advantages and disadvantages of Scrubbing mechanisms.

Scrubbing mechanisms		Advantages	Disadvantages
Blind scrubbing		-Corrects SBU and MBU	-Permanent availability of the Golden copy -High power consumption -Time overhead - area overhead
Readback scrubbing	Readback and Repair with ECC	-Detects and corrects SBU -Detects	-Complex -Time overhead
	Readback with		-High power

	CRC and Repair with the Golden copy	MBU	consumption -Time overhead -area overhead
	Readback with CRC and Repair with ECC		-Complex -Less time overhead
	Readback with CRC and Repair with the Golden copy		-Area overhead -Less power overhead
	Readback with CRC and Repair with partial reconfiguration		-Less time overhead

5 CONCLUSIONS

After defining the SRAM based configuration memory, the soft errors, the different faults tolerance methods and the advantages and disadvantages of each one of them. AS a conclusion of this review, we think that the most used and effective mitigation technique is scrubbing. We choose to adopt scrubbing, because with this technique, it is possible to achieve lower energy consumption, as the scrubbing is enabled only when a soft error is detected. In the mean while we are working on developing an optimized scrubbing approach to detected and correct faults caused by SEU, to improve the reliability of the SRAM based FPGA, in purpose of optimizing the attitude control system.

REFERENCES

Bouras, M., Berbia, H., Nasser, T., 2016. On Modeling and Fault Tolerance of NanoSatellite Attitude Control System. In *El Oualkadi A., Choubani F., El Moussati A. (eds) Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015*. Lecture Notes in Electrical Engineering, vol 380. Springer, Cham.

Colodro-Conde, C., & Toledo-Moreo, R., 2015. Design and analysis of efficient synthesis algorithms for EDAC functions in FPGAs. IEEE Transactions on Aerospace and Electronic Systems, 51, 3332-3347.

Dutta, A., Touba, N. A., 2007. Multiple Bit Upset Tolerant Memory Using a Selective Cycle Avoidance Based SEC-DED-DAEC Code. In *25th IEEE VLSI Test Symmposium (VTS'07),0-7695-2812-0/07.*

Xavier, X. J., Kantham, L., 2013. Multi Bit Upset Deduction/Correction for Memory Applications. International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868, Foundation of Computer Science FCS, New York, USA, Volume 5– No.3, February 2013.

Lankesh, M., Narasimhamurthy, K.C., 2015. Hardware Implementation of Single Bit Error Correction and Double Bit Error Detection through Selective Bit Placement for Memory. In *National Conference "Electronics, Signals, Communication and Optimization" (NCESCO 2015).* International Journal of Computer Applications (0975 – 8887)

Siegle, F., Vladimirova, T., 2015. Mitigation of Radiation Effects in SRAM-Based FPGAs. ACM Comput. Surv., vol. 47, no. 2, p. 37, Jan. 2015.

Brosser, F., Milh, E., Geijer, V., Larsson-Edefors, P., 2014. Assessing scrubbing techniques for Xilinx SRAM- based FPGAs in space applications. In *Proceedings of the 2014 International Conference on Field- Programmable Technology, FPT 2014*.

Xilinx, 2005. *Virtex-II Platform FPGA User Guide*, Xilinx and Inc. UG002 (v2.0) 23 March 2005.

Tonfat, J., Kastensmidt, F.G., Reis, R.A., 2015. Energy efficient frame-level redundancy scrubbing technique for SRAM-based FPGAs. In *2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 1-8*.

Legat, U., Biasizzo, A., Novak, F., 2012. SEU recovery mechanism for SRAM-Based FPGAs. IEEE Trans. Nucl. Sci., vol. 59, no. 5 PART 3, pp. 2562–2571, Oct. 2012.

Wirthlin, M., Harding, A., 2016. Hybrid Configuration Scrubbing for Xilinx 7-Series FPGAs. In *FPGAs and Parallel Architectures for Aerospace Applications, Cham*: Springer International Publishing, 2016, pp. 91–101.

Herrera-Alzu, I., López-Vallejo, M., 2013. Design techniques for Xilinx Virtex FPGA configuration memory scrubbers. IEEE Trans. Nucl. Sci., vol. 60, no. 1, 2013.

Berg, M., Poivey, C., Petrick, D., Espinosa, D., Lesea, A., LaBel, K. A., Friendlich, M., Kim, H., Phan, A., 2008. Effectiveness of internal versus external SEU scrubbing mitigation strategies in a Xilinx FPGA: Design, test, and analysis. In IEEE Transactions on Nuclear Science, 2008, vol. 55, no. 4, pp. 2259–2266.

Xilinx, 2010. *Xilinx XAPP864 SEU Strategies for Virtex-5 Devices, Application Note*, Xilinx and Inc.

Yang, J.-M., Kwak, S. W., 2015. Corrective control for transient faults with application to configuration controllers. IET Control Theory Appl., vol. 9, no. 8.