POLITECNICO DI TORINO
Repository ISTITUZIONALE

Analysis of Radiation-induced Cross Domain Errors in TMR Architectures on SRAM-based FPGAs

Original
Analysis of Radiation-induced Cross Domain Errors in TMR Architectures on SRAM-based FPGAs / Sterpone, Luca; Boragno, Luca. - ELETTRONICO. - (2017), pp. 174-179. (Intervento presentato al convegno International On-Line Testing and Robust System Design Symposium tenutosi a Thessaloniki nel July 3-5, 2017) [10.1109/IOLTS.2017.8046214].

Availability:
This version is available at: 11583/2680574 since: 2017-09-18T12:50:36Z

Publisher:
IEEE

Published
DOI:10.1109/IOLTS.2017.8046214

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

21 June 2024

Analysis of Radiation-induced Cross Domain Errors in TMR Architectures on SRAM-based FPGAs

Luca Sterpone, Luca Boragno
CAD Group –Dipartimento di Informatica e Automatica
Politecnico di Torino
Torino

***Abstract*—SRAM-Based FPGAs represent a low-cost alternative to ASIC device thanks to their high performance and design flexibility. In particular, for aerospace and avionics application fields, SRAM-based FPGAs are increasingly adopted for their configurability features making them a viable solution for long-time applications. However, these fields are characterized by a radiation environment that makes the technology extremely sensitive to radiation-induced Single Event Upsets (SEUs) in the SRAM-based FPGA's configuration memory. Configuration scrubbing and Triple Modular Redundancy (TMR) have been widely adopted in order to cope with SEU effects. However, modern FPGA devices are characterized by a heterogeneous routing resource distribution and a complex configuration memory mapping causing an increasing sensitivity to Cross Domain Errors affecting the TMR structure. In this paper we developed a new methodology to calculate the reliability of TMR architecture considering the intrinsic characteristics of the new generation of SRAM-based FPGAs. The method includes the analysis of the configuration bit sharing phenomena and of the routing long lines. We experimentally evaluate the method of various benchmark circuits evaluating the Mean Upset To Failure (MUTF). Finally, we used the results of the developed method to implement an improved design achieving 29x improvement of the MUTF.**

Keywords—SRAM-based FPGA; TMR; SEU; Cross Domain Errors; Reliability; Static Analysis,

I. INTRODUCTION

SRAM-based FPGAs offer a suitable solution in applications where flexibility and low implementation cost are the main goals. Moreover, their increasing performances make them an excellent alternative to more expensive ASIC solutions in space and avionics fields [1]. The main drawback in these fields is the harsh environment, which consists of radiation particles affecting the silicon area of electronic devices and generating Single Event Effects (SEEs) [2].

In particular, SRAM-based FPGAs store the configuration information in a volatile memory based on SRAM cells called Configuration RAM (CRAM). Radiation particle crossing the CRAM cell can produce a modification of the stored value. This phenomenon is known as Single Event Upsets (SEUs) and it can have unpredictable consequences on the implemented design. The SEU is not permanent but it can be fixed only at the refreshing of the values stored in the CRAM [3] through the scrubbing mechanism.

In order to mitigate these effects, two types of approaches have been proposed: methods based on scrubbing and on redundancy.

Configuration scrubbing consists in refreshing the content of the CRAM with error-free information in order to clean the memory from possible upsets and restore the circuit functionality. The scrubbing leads to correct multiple upsets but requires an external device to manage the reconfiguration. Moreover, it increases the power consumption and reduces the system availability during the refresh process [4] [5].

The most common redundancy technique is the Triplicated Modular Redundancy (TMR) where three domains are voted and works until two of them provide correct outcomes. The voting system masks the radiation effects propagation and the system continues its mission. However, the redundancy introduces an area overhead (more than 3.5 times the original circuit) and more in particular, TMR has a not predictable degradation when accumulated SEUs are considered [6], for this reason, it is usually combined with configuration scrubbing [7].

Cross Domain Errors (CDEs) shown in Fig. 1 are the main causes of TMR failure [8]. A CDE is a soft error that invalidates the protection capability of the TMR scheme and can be provoked by a SEU or because of Multiple Error Upsets (MEUs). In the first case, a SEU induced by a single particle (SEU-1p) affects a bit shared between two different domains [9]. In the second case, we have to distinguish between MEU induced by a single particle (MEU-1p) or provoked by multiple particles (MEU-mp). In MEU-1p the particle has sufficient energy to corrupt neighborhood SRAM cells of the configuration memory [10]. Vice versa, the MEU-mp is induced by several particle hits that are typically obtained after long time radiation exposure. This accumulation scenario is characterized by multiple memory cells modification in various locations [11].

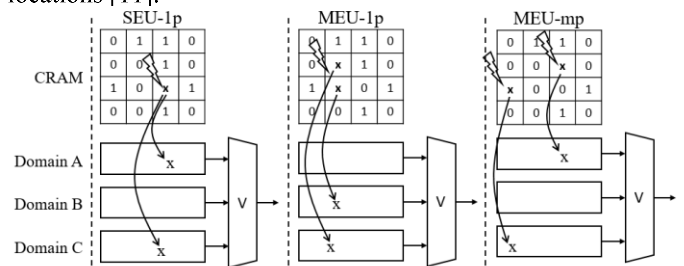

Fig. 1. The schemes of Cross-Domain Errors failures in TMR architecture on SRAM-based FPGAs.

The main contribution of this work consists on the fault tolerance analysis of different TMR benchmarks implemented on the new generation of SRAM-based FPGAs. We developed a new static analysis tool in order to detect the correlation between the circuit placement and its reliability. In particular, we computed the fault tolerance of a TMR architecture in terms of upsets to failure and we identified a consistent improvement by constraining the TMR logic placement with respect to the standard one obtained by means of commercial tools.

The paper is organized as follow: Section II gives an overview of the related works regarding the reliability of these devices; Section III describes the developed analysis tool parts such as routing database, configuration memory model, design modelization and static analyzer. Section IV provides a detailed experimental result analysis. Finally, Section V illustrates conclusions and discusses future steps.

II. RELATED WORKS

Several research studies focused on radiation effects analysis on TMR architectures implemented on SRAM-based FPGAs.

A redundant architecture is able to mask single soft-errors in flip-flops completely. However, harsh environments with high SEU rate lead to an accumulated scenario. This happen when a replica is corrupted by an upset and a second instance is disturbed before the faulty first one recovers in time. Since TMR does not handle faults affecting more than one module, in [12] a new fault tolerance solution limited to Look Up Table (LUT) is proposed. The research in [13] proposes a fine grain TMR at technological level in order to manage multiple upsets in nanoscale architectures. MEU detection technique based on parity bit computation is proposed in [14] while a multiple soft error tolerant platform combining redundancy and error-correction code is presented in [15].

Shared resources and common signals such as power supply and clock signals will affect the cores in the same way in case of fault occurrence. In [16] it has been demonstrated that pulses in the power supply cause timing violation in the critical path. Error related to wrong signal sampling in TMR is faced in [17]. In particular authors propose TMR synchronizers in order to counteract the effects of asynchronous sampling of cross-clock domain signals.

The TMR implementation intrinsically defines its reliability. Previous analysis performed in [18] highlights that minimizing the wire length, and thus the number of routing resources reduces the existence of SEU sensitive nets. In the same work we calculated that 72% of all the configuration memory bits controlling a switch box could produce critical situations if used for routing different TMR replicas. In [19], we implemented different versions of the same redundant circuit for radiation test. Considering the classification of the type of interconnection in the design, we identified a correlation between the amount of independent routes in the design and the reliability enhancement of the redundant versions.

In this work, we extend the analysis regarding the interaction between the kind of routing resources employed and their impact on the design reliability. In particular, we focus on the long lines behavior and the shared bit effect.

III. THE DEVELOPED METHOD

The developed method consists on the flow illustrated in Fig. 2. The analysis method elaborates a circuit implemented on a SRAM-based FPGA processing it versus a configuration memory model, which reproduces the effective logic and routing resources used in the design implementation. A routing database containing architectural information of shared resources and long line has been developed and used for instrumenting the Configuration Bit (CB) mapping, while the circuit description is exported in a custom format. The static analyzer performs the reliability computation in terms of Sensitive bits and Mean Upset To Failure (MUTF).

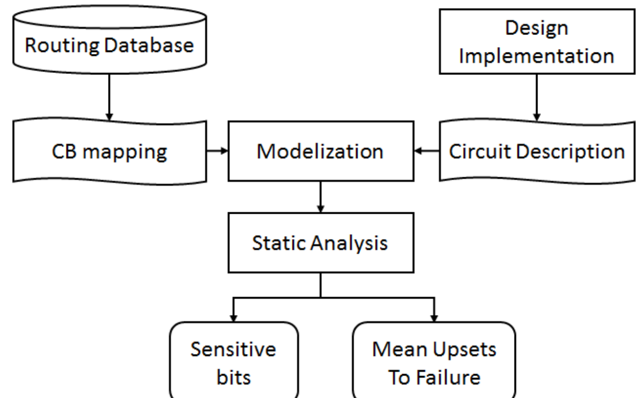

Fig. 2. The flow of the developed analysis method.

A. *Background*

The sharing of a CB between two different domains is a critical situation that can produce a CDE and lead to a TMR failure. As depicted in Fig. 3, there are several situations in a switch box (SB), where one SEU induces multiple errors violating the single fault assumption [20]. In particular, an upset in the configuration frame of the SB changes the connection layout, producing a conflict between two nets and unpredictable behaviors in the carried logic value.

Besides, is not necessary that two lines of different domains share the same SB to get in conflict. Long lines have been considered as a source of CDEs. They carry signals across the length or width of the chip with minimal delay and negligible skew, which are connected to a primary global net or to any secondary global net. Long lines are classified as Horizontal Long Line (HLL) and Vertical Long Line (VLL). Considering a SB in a tile (X,Y), they allow a fast connection to another tile far n on the vertical axis or distant m on the horizontal axis. The parameters m, n are strictly related to the device under the study.

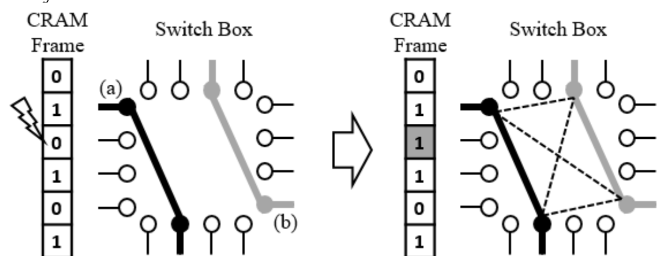

Fig. 3. The corrupted CB belongs to the SB shared between the net (a) and net (b) which appertain to different domains. Its modification actives undesired connection (dashed lines) with a TMR failure.

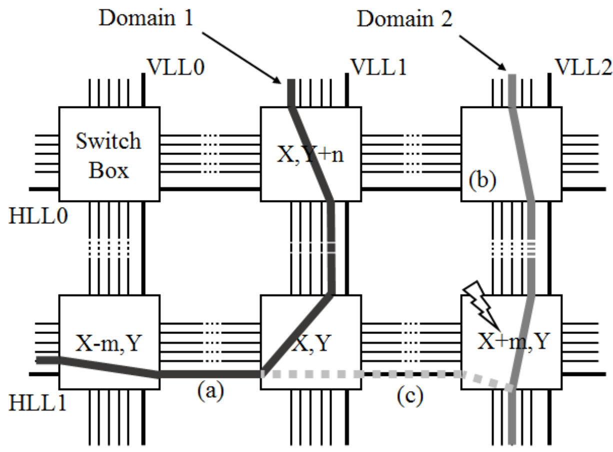

Fig. 4. Consider net (a) and net (b) belonging to two different domains, and net (a) uses partially the HLL1 from (X-m,Y) to (X,Y). An upset in (X+m,Y) can create a link net (c) connecting two different domains.

Considering a fault in the SB (X,Y+m) used in the domain "1" which activates a connection to the long line used in (X,Y) by the domain "2", a conflict failure between two different domains due to a long line has been created. This CDE effect is illustrated with the FPGA layout drafted in Fig. 4.

B. Analysis Tool

The reliability analyzer uses a parametric configuration memory model that can be adjusted in order to recreate the CRAM of different FPGAs. The tool needs a description of the circuit containing information regarding the principal resource utilization, Look-Up-Table (LUT), Flip-Flop (FF), Programmable Interconnection Point (PIP). Additionally, we built a database with the information about the CBs for each pip. We developed a script to extract the circuit netlist in an internal format in order to be compliant with our database and be processable by the static analyzer. Moreover, the developed tool is able to map the design on its configuration memory model, to perform a preliminary fault analysis and to estimate the design susceptibility.

C. Database

The database contains information about the CBs for each PIP typology. Due to the heterogeneous layout, an FPGA has several tiles, which allow the communication with special blocks such as DSP, BRAM, Clock resources and with the boundaries resources like IO blocks or Transceivers IO. Nevertheless, is possible to retrieve regular patterns in the PIP structure replicated in many locations inside the device. The database collects the information of these unique patterns, defined *individual PIP*. A database of individual PIPs is 1400x smaller than a database of all the FPGA interconnection points, thus enabling a feasible analysis. The built database holds for each individual PIP the kind of the belonging tile and the set of CBs. Each CB is identified by a *bit index*, which is an offset from a custom origin. From the location coordinates and the bit index is possible map all the FPGA's PIPs in the configuration memory model.

D. Configuration memory model

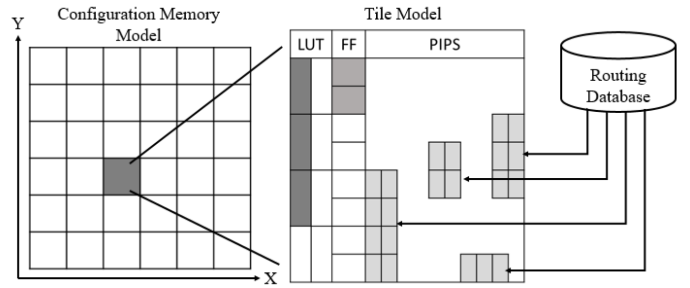

Fig. 5. The configuration memory model is a matrix of tile. Each tile has a fixed amount of CBs for LUT, FF and PIPs.

The parametric configuration memory model developed is illustrated in Fig. 5. The model is able to define the matrix of tiles contained in the whole FPGA architecture.

Each tile is divided in three parts: LUT, FF, and PIP parts. As it has been mentioned, it is completely parametric. Therefore, the Configuration Word (CW) for each resource and the amount of resources per tile is customizable. For what concern LUTs and FFs, the CW length is equal to all. In case of PIPs instead, this quantity changes in agreement with the connection type. The number of CBs required is retrieved from the routing database as well as the bit index.

The tool traces how many times a CB is marked as used. The same as considering long line PIPs. It sets sensitive not only the bits in the tile under analysis but also in the tiles that the long line connects, as shown in Fig. 6.

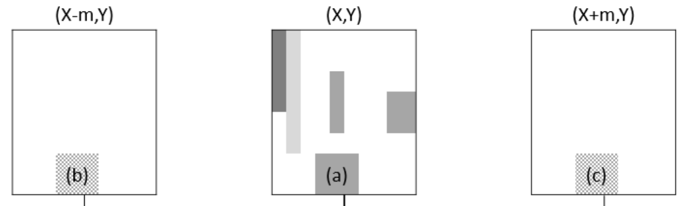

Fig. 6. An overview of the configuration memory layer considering long line topology: when, in the tile (X,Y), the set of bits (a) configuring a long line PIP is used, the correspondent set of bits (b) and (c) in the tiles (X±m,Y) are set sensitive.

E. Circuit modelization

The circuit modelization process uses the resource location data to select the tile where lay the corresponding CW in our memory model. Place a resource consist in set as used the corresponding CBs, determine the domain of belonging, and define the used time.

In order to consider the bit sharing, during the modelization, if a location, which has been used by another resource, is found, and that resource belongs to a different domain, the location is set as critical.

The same as considering the long line effects. When a PIP belongs to a long line, the process sets as sensitive also the PIPs in the connected tiles. If this operation finds a bit already used in a different domain, the configuration is set as critical. This process has been implemented in an algorithm, as illustrated in the pseudo-code in Fig. 7.

```
/*Initialization*/
readDatabase()
importDesign()
/*Placement*/
foreach lut
 setUsed(lut.x,lut.y,domain)
foreach ff
 setUsed(ff.x,ff.y,domain)
foreach pip
/* shared bit implementation */
 if isUsed() == false
  setUsed(pip.x,pip.y,domain)
 else
  if  sameDomain() == false
   setAsCritical()
  else
   setUsed(pip.x,pip.y,domain)
/* long line implementation*/
 if  isLongLinePip() == true
  if  sameDomain(offset) == false
   setAsCritical()
  else
   setUsed(pip.x,pip.y,domain)
```

Fig. 7. The pseudo-code of the circuit modelization algorithm able to handle the shared bits and the long lines behavior.

F. Reliability static analyzer

The reliability analyzer aims to compute the circuit fault tolerance at the design phase without any additional hardware. In particular, it performs a static analysis over the configuration memory in order to individuate the sensitive bits of the circuit implemented on the SRAM-based FPGA. The classification has been defined considering as a faulty condition, a configuration matrix including one or more sensitive bits. In order to be compliant with TMR architecture, redundant designs fail when two of the three configuration domains are faulty.

The developed analysis method specifically focuses on the effect of a long radiation exposition and extract a meaningful statistic related to a radiation test or realistic exposure. The developed analysis method is able to inject faults and to perform multiple evaluations of SEUs, recreating an accumulation scenario. Moreover, it is able to perform multiple runs of the same scenario to refine on the results. The tile and the bit to inject is computed randomly. For each analysis scenario (with single or multiple SEUs), the tool returns the amount of sensitive bits in the design, i.e. the bits which corruption can lead to a potential system failure.

The reliability computation on the Mean Upset to Failure (MUTF) has been performed [21]. The MUTF parameter represents the average amount of accumulated SEUs in the configuration memory that lead to a failure. It is computed as the ratio between the number of faults injected in each design (n) and the number of observed failures on the design (k):

$$MUTF = \frac{n}{k} \quad (1)$$

The system continuously injects the configuration memory model until detects a faulty configuration. To continue, it records the number of upsets injected and repeats the flow for a user-defined amount of iterations.

The design sensitivity is the percentage of CBs within the FPGA design that are sensitive to upsets. Using the maximum likelihood estimator r, of the Binomial distribution:

$$r = k / n \quad (2)$$

The standard deviation of the maximum likelihood estimator is:

$$\sigma = \sqrt{\frac{k}{n^2}\left(1 - \frac{k}{n}\right)} \quad (3)$$

The standard deviation of the estimator can be used to determine the 95% confidence interval bounds of the sensitivity estimated, i.e. $\pm 2\sigma$. The number of sensitive bits variation is estimated for each design by multiplying the total number of CRAM bits in the device (e.g., Xilinx Kintex-7 XC7K325T counts up to 91,548,896 configuration memory bits) by the sensitivity of the design. Designs with lower sensitive bits require much more upsets to get faulty configuration.

IV. Experimental Results

We implemented different circuits on a XC7K325T Kintex-7 FPGA embedded in the KC705 Xilinx evaluation kit. The reliability analyzer was tuned according to the Kintex-7 parameters for what concern dimensions, configuration words, logic and routing resources counts.

A. Benchmarks

In order to analyze the effects of shared bits and long lines on the design reliability, we studied the dependency between the topology placement and the error rate characteristic of a circuit implementation. We developed different circuit benchmarks in plain and TMR architecture of a LEON3 and other set of circuit. Considering the ITC'99 benchmarks, we selected B11, B14 and B15; while from ISCAS85 we used C499 and C6288, while as microprocessor case study we selected the Leon3 processor. Table I reports the implementation results regarding the LUT and the FF utilization.

Table I. Implementation area results for the circuits benchmarks in plain and TMR version.

Benchmark Circuit	LUT [#]		FF [#]	
	plain	*TMR*	*plain*	*TMR*
b11	108	324	30	90
b14	2,350	6,882	218	654
b15	2,295	6,918	418	1,251
c499	63	192	-	-
c6288	686	2,061	-	-
Leon3	4,439	13,317	1,950	5,850

B. Reliability analysis

We performed different reliability analysis over all the circuit benchmarks on its plain and TMR versions. Both of the implementations are obtained with the standard Synthesis Strategy of the Xilinx Vivado Design Suite v2015.4.

Table II. Detailed Experimental results

		Fault Injected (n)	Failures (k)	MUTF	Sensitivity [%]	Sensitive Bits	Std.Dev. [%]
b11	**Plain**	290,306	986	294	0.340	310,938	0.011
	TMR	1,42,0941	524	2,712	0.037	33,760	0.002
	TMR_AG	3,002,583	812	3,698	0.027	24,758	0.001
b14	**Plain**	160,806	20,067	8	12.48	11,424,398	0,082
	TMR	11,646,271	90,183	129	0.774	708,910	0.003
	TMR_AG	15,054381	111,319	135	0.739	676,955	0.002
b15	**Plain**	317,532	39,844	8	12.55	11,487,580	0.059
	TMR	15,014,237	132,261	114	0.881	806,458	0.002
	TMR_AG	12,879,649	96,155	134	0.747	683,472	0.002
c499	**Plain**	90,924	694	131	0.763	698,770	0.029
	TMR	3,862,589	5,637	685	0.146	133,605	0.002
	TMR_AG	585,858	222	2,639	0.038	34,691	0.002
c6288	**Plain**	16,286	958	17	5.882	5,385,229	0.184
	TMR	11,499,389	2,9129	395	0.253	231,902	0.001
	TMR_AG	3,092,460	6,265	494	0.203	185,468	0.003
Leon3	**Plain**	252,678	56,458	4	22.34	20,455,550	0.083
	TMR	10,835,640	193,281	56	1.784	1,633,006	0.004
	TMR_AG	11,081,737	188,526	59	1.701	1,557,459	0.004

An additional TMR implementation with area group is carried out called TMR_AG. This version has been implemented in order to avoid cross-domain errors caused by the bit sharing between different replicas, this adjustment aims to make the domains as much isolated as possible as shown in Fig. 8. This implementation requires the same area resources of the previous one. Furthermore, from the timing analysis, we observed in this version an overall performance improvement, in term of maximum frequency achievable.

The Fig. 9 reports the utilization of PIPs belonging to long lines. Triplicated designs employ much more long lines with respect to the plain versions. Their utilization could be reduced with a different placement policy as obtained in the area group versions.

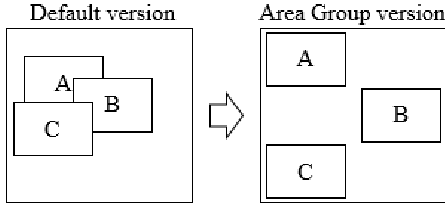

Fig. 8. The area groups in TMR_AG avoid the domain interleaving.

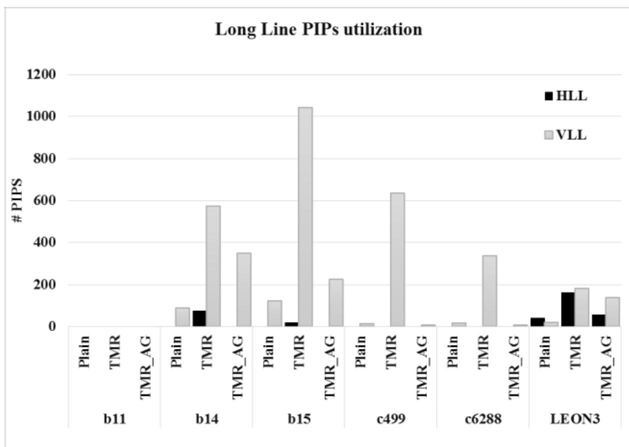

Fig. 9. Long line PIPs usage comparison among the circuit implementations.

Regarding the bit sharing, the histogram illustrated in Fig. 10 reports the critical bits in the analyzed circuits. A critical bit is a bit whose corruption lead to a CDE. Critical Shared Bits (CSBs) are bits shared among resources belonging to different domains. Critical Long Line Bits (CLLBs) instead, are sensitive bits that can create a connection between two domains due to the long line effect. Critical Shared Long Line Bits (CSLLBs) are shared among domains and potential cause of long line effects. It is possible to notice that by avoiding the interleaving between TMR modules we reduced significantly the number of bits able to produce a CDE.

Finally, in Table III are reported the obtained improvement of MUTF in the designs under analysis. With a different placement policy, we increase the MUTF up to 29x with respect a plain version. Considering MUTF, the impact of shared bits and long lines utilization in the system reliability is efficiently recognizable. Detailed experimental results obtained from 1,000,000 iterations are reported in Table II. In particular, designs having fewer sensitive bits require much more fault injected to get good results, i.e. with the 95% of confidence.

Considering the LEON3 processor, we observed an improvement of 4x in the TMR implementation. Formerly, enhance this value to 27x with the combination of TMR and CRAM scrubbing that can be increased up 50x adding BRAM scrubbing [21]. All results have a 95% of level confidence.

With our area group placement, we are able to increase up to 15x the MUTF without any scrubbing cycle. Increasing the MUTF implies a lower amount of scrubbing cycles, i.e. less power consumption and higher module availability. Availability becomes crucial in real time applications.

Table III. MUTF improvement comparison

	MUTF Improvements		
	Plain	**TMR**	**TMR_AG**
b11	1x	9x	13x
b14	1x	16x	17x
b15	1x	14x	17x
c499	1x	5x	20x
c6288	1x	23x	29x
Leon3	1x	14x	15x

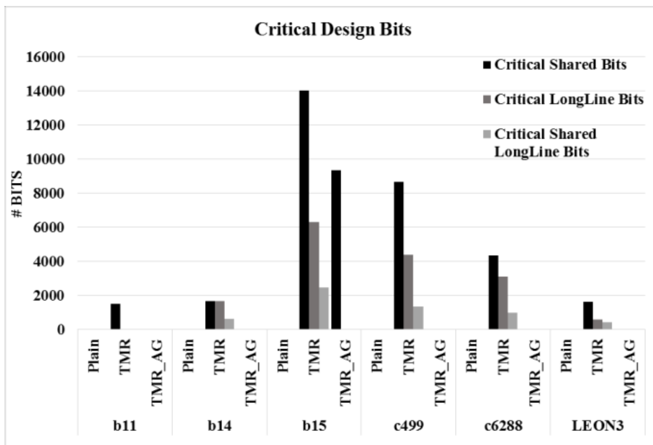

Fig. 10. Critical design bits in all the circuit versions. Bits are divided in CSBs, CLLBs and CSLLB.

V. Conclusion and Future Works

In this work we investigated the sources of Cross Domain Errors in redundant architectures that invalidate the single-fault assumption. In particular, the configuration bit sharing among resources and the impact of long lines utilization in the design have been studied. We built a database with the configuration memory bits for each PIPs for the Kintex-7 FPGA family and we developed a configuration memory model and a custom circuit description in order to perform static analysis on the design reliability. These studies have been carried out on different circuits coming from ITC99, ISCAS85 benchmarks and LEON3 processor. Different implementation strategies of the same circuit has been tested. The obtained results shown an improvement up to x29 the original version in MUTF parameter thanks to an area group placement focused on the bit sharing and long line usage reduction.

As future works, we plan to refine on the configuration memory model introducing additional heterogeneous FPGA features such as DSP, BRAM and other type of tiles. In addition we will define an area group policy and a placement metric for TMR systems. Finally we plan to extend the long lines effect definition to the local lines and test other source of CDEs.

References

[1] N. Montealegre, D. Merodio, A. Fernández and P. Armbruster, "In-flight reconfigurable FPGA-based space systems," *2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, Montreal, QC, 2015, pp. 1-8.s

[2] N. Bidokhti, "SEU concept to reality (allocation, prediction, mitigation)," *2010 Proceedings - Annual Reliability and Maintainability Symposium (RAMS)*, San Jose, CA, 2010, pp. 1-5.

[3] R. H. Maurer, M. E. Fraeman, M. N. Martin, D. R. Roth, "Harsh Environments: Space Radiation Environment Effects and Mitigation", *Johns Hopkins APL Technical Digest*, vol. 28, 2008.

[4] A. Nafkha and Y. Louet, "Accurate measurement of power consumption overhead during FPGA dynamic partial reconfiguration," *2016 International Symposium on Wireless Communication Systems (ISWCS),* Poznan, Poland, 2016, pp. 586-591.

[5] L. Sterpone, L. Boragno and D. M. Codinachs, "Analysis of radiation-induced SEUs on dynamic reconfigurable systems," *2016 11th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC),* Tallinn, 2016, pp. 1-6.

[6] G. Foucard, P. Peronnard and R. Velazco, "Reliability limits of TMR implemented in a SRAM-based FPGA: Heavy ion measures vs. fault injection predictions," *2010 11th Latin American Test Workshop, Pule del Este*, 2010, pp. 1-5.

[7] Aitzan Sari, Mihalis Psarakis, Dimitris Gizopoulos, "Combining checkpointing and scrubbing in FPGA-based real-time systems", *IEEE 31st VLSI Test Symposium*, 2013, pp 1-6.

[8] H. Quinn, K. Morgan, P. Graham, J. Krone, M. Caffrey and K. Lundgreen, "Domain Crossing Errors: Limitations on Single Device Triple-Modular Redundancy Circuits in Xilinx FPGAs," *IEEE Transactions on Nuclear Scienc*e, vol. 54, no. 6, pp. 2037-2043, Dec. 2007.

[9] M. Ceschia, M. Violante, M. Sonza Reorda, A. Paccagnella, P. Bernardi, M. Rebaudengo, D. Bortolato, M. Bellato, P. Zambolin, and A. Candelori, "Identification and Classification of Single-Event Upsets in the Configuration Memory of SRAM-Based FPGAs," *IEEE Trans. Nuclear Science*, vol. 50, no. 6, pp. 2088-2094, Dec. 2003.

[10] H. Quinn, P. Graham, J. Krone, M. Caffrey, S. Rezgui, "Radiation-induced multi-bit upsets in SRAM-based FPGAs", *IEEE Transactions on Nuclear Science*, Vol. 52, Issue 6, pp. 2455 – 2461, 2005.

[11] H. Abbasitabar, H. R. Zarandi and R. Salamat, "Susceptibility Analysis of LEON3 Embedded Processor against Multiple Event Transients and Upsets," *2012 IEEE 15th International Conference on Computational Science and Engineering*, Nicosia, 2012, pp. 548-553.

[12] C. Argyrides, H. Zarandi and D. K. Pradhan, "Multiple SEU tolerance in LUTs of FPGAs using protected schemes," *2008 European Conference on Radiation and Its Effects on Components and Systems*, Jyvaskyla, 2008, pp. 325-330.

[13] M. Niknahad, O. Sander and J. Becker, "FGTMR - Fine grain redundancy method for reconfigurable architectures under high failure rates," *The 16th North-East Asia Symposium on Nano, Information Technology and Reliability*, Macao, 2011, pp. 186-191.

[14] S. Aishwarya and G. Mahendran, "Multiple bit upset correction in SRAM based FPGA using Mutation and Erasure codes," *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, Ramanathapuram, 2016, pp. 202-206.

[15] M. Amagasaki, Y. Nakamura, T. Teraoka, M. Iida and T. Sueyoshi, "A novel soft error tolerant FPGA architecture," *2016 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Tallinn, 2016, pp. 1-6.

[16] P. Tummeltshammer and A. Steininger, "On the role of the power supply as an entry for common cause faults—An experimental analysis," *2009 12th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, Liberec, 2009, pp. 152-157.

[17] Y. Li, B. Nelson and M. Wirthlin, "Synchronization Techniques for Crossing Multiple Clock Domains in FPGA-Based TMR Circuits," *IEEE Transactions on Nuclear Science*, vol. 57, no. 6, pp. 3506-3514, Dec. 2010.

[18] L. Sterpone and M. Violante, "A new reliability-oriented place and route algorithm for SRAM-based FPGAs," *IEEE Transactions on Computers*, vol. 55, no. 6, pp. 732-744, June 2006.

[19] Boyang Du, L. Sterpone, L. Venditti and D. M. Codinachs, "On the design of highly reliable system-on-chip using dynamically reconfigurable FPGAs," *2015 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, Bremen, 2015, pp. 1-6.

[20] L. Sterpone, G. Cabodi, S. F. Finocchiaro, C. Loiacono, F. Savarese and B. Du, "Scalable FPGA graph model to detect routing faults," *2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, Sant Feliu de Guixols, 2016, pp. 155-160.

[21] A. M. Keller and M. J. Wirthlin, "Benefits of Complementary SEU Mitigation for the LEON3 Soft Processor on SRAM-Based FPGAs," *IEEE Transactions on Nuclear Science*, vol. 64, no. 1, pp. 519-528, Jan. 2017