UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

ANTONIO FELIPE COSTA DE ALMEIDA

Investigating Techniques to Reduce Soft Error Rate under Single-Event-induced Charge Sharing

Thesis presented in partial fulfillment of the requirements for the degree of Master of Microelectronics.

Advisor: Prof. Dr. Fernanda Lima Kastensmidt

Porto Alegre
2014

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Almeida, Antonio Felipe Costa de

Investigating Techniques to Reduce Soft Error Rate under Charge Sharing / Felipe Almeida. – Porto Alegre: PGMICRO da UFRGS, 2014.

84 f.:il.

Advisor: Fernanda Lima Kastensmidt.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2013.

1. Fault Tolerance Techniques. 2. Placement Constraints Methodology. 3. Multiple Faults Scenario. I. Kastensmidt, Fernanda Lima. II. Investigating Techniques to Reduce Soft Error Rate under Single-Event-induced Charge Sharing.

AGRADECIMENTOS

E é dada como terminada mais uma etapa da minha vida. Demorou, mas consegui concluir, mesmo com toda a correria de tentar fazer um mestrado e trabalhar ao mesmo. Guardo boas lembranças desse período aqui, foram vários artigos aceitos, bons congressos, sem contar com as companhias para as viagens. Só tenho a agradecer a todos que me ajudaram e fizeram parte desta dissertação.

Primeiramente, queria agradecer a minha família, pois sem o suporte deles, amor e coragem não teria chegado aqui. A minha mãe, Zefinha, meu pai, Jaime e minha irmã, Fernanda. Essa educação que recebi em casa, a faculdade e ninguém poderia ter me dado melhor. Aos que ajudaram nesse processo de formação e ainda guardo seus ensinamentos, como: Irene, minha tia, Ivanilda, tia postiça, Gilva, professora e Adessandro, técnico de natação.

Aos amigos da UFRGS que estiveram comigo nas disciplinas, tirando dúvidas no laboratório e na loucura de conseguir terminar os artigos no prazo. São eles: Samuel, Eduardo Souza, Eduardo Chielli, Lucas, Anelise, Carol, Gracieli, José Rodrigo, Jimmy, Jorge e William.

A orientadora e amiga, Fernanda. Essa merece mais do que estar nos agradecimentos. Ela foi responsável por toda a concretização dos artigos aceitos em congressos e revistas. Bem como, me guiou na escolha do tema, direcionou a pesquisa e ajudou na escrita deste e outros textos. Primeira etapa concluída, agora vamos para o doutorado, que já foi iniciado e pelo menos mais 3 anos de parceria, muito obrigado por tudo.

Acredito que não se consiga fazer nada, por mais conhecimento que se tenha, se não existe o apoio dos amigos que me acompanham até hoje. Não são muitos, mas os que ficaram ao meu lado são os melhores do mundo. Quero agradecer em especial aos meus amigos de Belo Horizonte, representado aqui pelo Carlos Junior e Diego Magela, aos amigos de São Paulo, por Daniel Vita e ao pessoal do vôlei, pela Meus SAIS. Os amigos de infância que fazem tanta falta Ludmilla e Michelle, bem como o pessoal da 8º série, represetando por Handricka. Os de Porto Alegre, os quais já chamo de família: Wendyel, Guile, Luis, Tanira, Paula e Regina (Bah).

Muitos outros me deram forças em determinadas fases da minha vida, fica aqui meu agradecimento por todos esses que tiveram um papel importante por um determinado tempo. Cada um ajudou e contribuiu da sua forma. Muito obrigado!

“I want to say I live each day, until I die
And all that I had something in, somebody's life
The hearts I had touched will be the proof that I leave
That I made a difference and this world will see”
B.K.

ABSTRACT

The interaction of radiation with integrated circuits can provoke transient faults due to the deposit of charge in sensitive nodes of transistors. Because of the decrease the size in the process technology, charge sharing between transistors placed close to each other has been more and more observed. This phenomenon can lead to multiple transient faults. Therefore, it is important to analyze the effect of multiple transient faults in integrated circuits and investigate mitigation techniques able to cope with multiple faults.

This work investigates the effect known as single-event-induced charge sharing in integrated circuits. Two main techniques are analyzed to cope with this effect. First, a placement constraint methodology is proposed. This technique uses placement constraints in standard cell based circuits. The objective is to achieve a layout for which the Soft-Error Rate (SER) due charge shared at adjacent cell is reduced. A set of fault injection was performed and the results show that the SER can be minimized due to single-event-induced charge sharing in according to the layout structure. Results show that by using placement constraint, it is possible to reduce the error rate from 12.85% to 10.63% due double faults.

Second, Triple Modular Redundancy (TMR) schemes with different levels of granularities limited by majority voters are analyzed under multiple faults. The TMR versions are implemented using a standard design flow based on a traditional commercial standard cell library. An extensive fault injection campaign is then performed in order to verify the soft-error rate due to single-event-induced charge sharing in multiple nodes. Results show that the proposed methodology becomes crucial to find the best trade-off in area, performance and soft-error rate when TMR designs are considered under multiple upsets. Results have been evaluated in a case-study circuit Advanced Encryption Standard (AES), synthesized to 90nm Application Specific Integrated Circuit (ASIC) library, and they show that combining the two techniques, the error rate resulted from multiple faults can be minimized or masked.

By using TMR with different granularities and placement constraint methodology, it is possible to reduce the error rate from 11.06% to 0.00% for double faults. A detailed study of triple, four and five multiple faults combining both techniques are also described.

We also tested the TMR with different granularities in SRAM-based FPGA platform. Results show that the versions with a fine grain scheme (FGTMR) were more effectiveness in masking multiple faults, similarly to results observed in the ASICs.

In summary, the main contribution of this master thesis is the investigation of charge sharing effects in ASICs and the use of a combination of techniques based on TMR redundancy and placement to improve the tolerance under multiple faults.

Keywords — **Fault tolerance, Triple Modular Redundancy, Single-Event-Induced Charge Sharing, Placement Constraining**

INVESTIGANDO TÉCNICAS PARA REDUZIR A TAXA DE ERRO DE SOFT SOB EVENTO ÚNICO INDUZIDO DE CARGA COMPARTILHADA

RESUMO

A interação da radiação com circuitos integrados pode provocar falhas transitórias devido ao deposito de cargas em nós sensíveis de transistores. Por causa da diminuição das dimensões no processo tecnológico, cargas compartilhadas entre trasistores posicionados próximos uns dos outros tem sido cada vez mais observadas. Este fenômeno pode causar múltiplas falhas transientes. Por isso, é importante analisar o efeito de múltiplas falhas transitórias em circuitos integrados e investigar técnicas de mitigação.

Este trabalho investiga o efeito conhecido como evento único induzido de carga compartilhada em circuitos integrados. Duas técnicas são analisadas para lidar com este efeito. Primeiro, uma técnica que utiliza restrições de posicionamento em circuitos baseados em células padrões é proposta. O objetivo é conseguir um leiaute para que a taxa de erro de soft (SER), devido ao compartilhamento de cargas em células adjacentes, seja reduzida. Um conjunto de injeção de falhas foi realizado e os resultados mostram que o SER pode ser minimizado com o leiaute devido ao evento único induzido de carga compartilhada. Resultados mostraram que pelo uso de restrições de posicionamento é possível reduzir a taxa de erro de 12,85% para 10,63% devido a falhas duplas.

Em segundo lugar, esquemas de redundância modular tripla (TMR) com diferentes níveis de granularidade limitados pelos votadores são analisados sob múltiplas falhas. As versões de TMR são implementadas usando um fluxo de projeto padrão com base em uma biblioteca de células padrão comercial e tradicional. Uma ampla campanha de injeção de falhas é então realizada, a fim de verificar a taxa de erro de soft devido ao evento único induzido de carga compartilhada em vários nós. Os resultados mostram que a metodologia proposta é crucial para encontrar o melhor custo-benefício entre área, desempenho e a taxa de erro de software ao considerar projetos de TMR sob múltiplas falhas. Resultados tem sido avaliados em um circuito de estudo de caso do Padrão Avançado de Criptografia, sintetizado para uma biblioteca de aplicação específica de circuitos integrados de 90nm e eles mostraram que a combinação das duas técnicas, a taxa de erro resultante de múltiplas falhas pode ser minimizado ou mascarado.

Por usar TMR com diferentes granularidades e metodologia de restrição de posicionamento é possível reduzir a taxa de erro de 11,06% para 0,00% para falhas duplas. Um estudo detalhado de múltiplas falhas: triplas, quádruplas e quíntuplas combinando ambas as técnicas também são descritas.

Nós também testamos o TMR com diferentes granularidades em plataformas de FPGA baseadas em SRAM. Os resultados mostram que a versão com um esquema de granularização fina (FGTMR) foi mais eficaz no mascaramento de múltiplas falhas, que são similares nos resultados observados em ASICs.

Em resumo, a principal contribuição desta dissertação é a investigação dos efeitos de cargas compartilhadas em ASICs e o uso da combinação das técnicas baseada em redundância de TMR e posicionamento para melhorar a tolerância sob múltiplas falhas.

Palavras-Chave — **Tolerância a falhas, Redundância Modular, Evento Único Induzido de Carga Compartilhada, Restrição de Posicionamento**

LIST OF ABBREVIATION AND ACRONYMS

AES	Advanced Encryption Standard
AMUSE	Autonomous Multilevel Emulation-Based for Soft Error Evaluation
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
CAD	Computer Aided Design
CGTMR	Coarse Grain Triple Modular Redundancy
CMOS	Complementary Metal-Oxide-Semiconductor
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CTS	Clock Tree Constraint
DDR	Design Diverse Redundancy
DEF	Design Exchange Format
DRAM	Dynamic Random Access Memory
DTMR	Diverse Triple Modular Redundancy
ESF	Error Status Flag
FGTMR	Fine Grain Triple Modular Redundancy
FIT	Failures in Time
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
HDL	Hardware Description Language
IC	Integrated Circuit
IMEC	Interuniversity Microelectronics Centre
LANL	Los Alamos National Laboratory’s
LANSCE	Los Alamos National Neutron Sience Center
LEF	Library Exchange Format
LET	Linear Energy Transfer
LETth	LET Threshold
MET	Multiple Event Transient
MEU	Multiple Bit Upset
MOS	Metal-Oxide-Semiconductor
MSET	Multiple Single Event Transient
MV	Majority Voter
SEE	Single-Event Effect

NASA	National Aeronautic and Space Administration
NMF	Non-Masked Fault
NMR	N-Modular Redundancy
RAD	Radiation Absorbed Dose
RISC	Reduced Instruction Set Computing
SAv	Self-Adapted Voter
SDC	Synopsys Design Constraint
SEB	Single Event Burnout
SEGR	Single Event Gate Rupture
SEL	Single Event Latch up
SER	Soft-Error Rate
SET	Single Event Transient
SEU	Single Event Upset
SI	International System of Units
SI	Signal Integrity
SPEF	Standard Parasitic Exchange Format
SRAM	Static Random Access Memory
STA	Static Timing Analysis
TID	Total Ionization Dose
TRM	Triple Modular Redundancy
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VHSIC	Very High Speed Integrated Circuit

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Some radiation effects on semiconductor devices have increased substantially with the decreasing of transistor size and the increasing amount of components for a given circuit area. Nowadays, it is possible to find integrated circuits fabricated in reduced dimensions such as 22 nanometers CMOS process technique operating in aerospace environments (BOHR, 2011). In this context, one of the most critical challenges in design is the amount of faults caused by cosmic ray particles. The radiation effects must be considered at the design phase to guarantee the high-reliability and safety requirements of such project (VELAZCO, 2007). Hence, it is necessary to implement techniques to avoid any interference that may cause malfunction on the system.

Integrated circuits operating in radiation environment and on Earth are susceptible to transient upsets caused by energetic particles. An upset can happen in a specific point, being called single fault, or in more than one point, called multiple faults. According to the amount of charge deposited by the energetic particle colliding with the silicon, transient pulse can have different shapes. Not all transient faults are going to provoke an error in the circuit. An error is any deviation from the expected behavior of a circuit or system. Transient upsets or faults may occur in memory elements or they may propagate through the combinational logic and if not masked by the logic or the application, these transient faults may lead to errors in the circuit.

Many of the multiple faults are due to single-induced charge sharing upsets, which means that a single energetic particle has deposited enough energy to perturb multiple transistors placed close to each other. This phenomenon is one of the most common causes of multiple upsets in nanometer technology (AMUSAN, 2006). Consequently, one of the main goals of this work is to investigate and develop a methodology to characterize integrated circuits under multiple faults by determining the most sensitive nodes and gates for fault mitigation.

In order to characterize circuits under single-induced charge sharing upsets it is mandatory to know the placement of the logic gates and transistors, because the multiple faults must be injected in transistors that are placed close to each other in the real circuit. Therefore, we have used and improved a tool, called Autonomous Multilevel Emulation-Based for Soft Error Evaluation (AMUSE) (ENTRENA, 2009). It was developed by Universidad Carlos III del Madrid (UC3M) under the guidance of the Dr. Luis Entrena. The tool is used to perform multiple fault injection and evaluate the Soft Error Rate (SER).

Once the susceptibility of a circuit under multiple faults is analyzed, some mitigation techniques can be investigated, verified and improved. A mitigation technique, which can be used to reduce charge sharing, is the placement constraint methodology (ENTRENA, 2012). Placing logic gates far away from each other, in order to reduce the error rate. This technique is based on the charge-sharing characteristics of the circuit layout.

The design flow can also be changed to introduce fault tolerance techniques that decrease the number of multiple faults. One modification is applied during the floorplan phase; it changes the utilization area, then it increases the layout space between cells as can be seen in figure 1. However, this approach has drawbacks: increase of area, routing, and circuit performance.

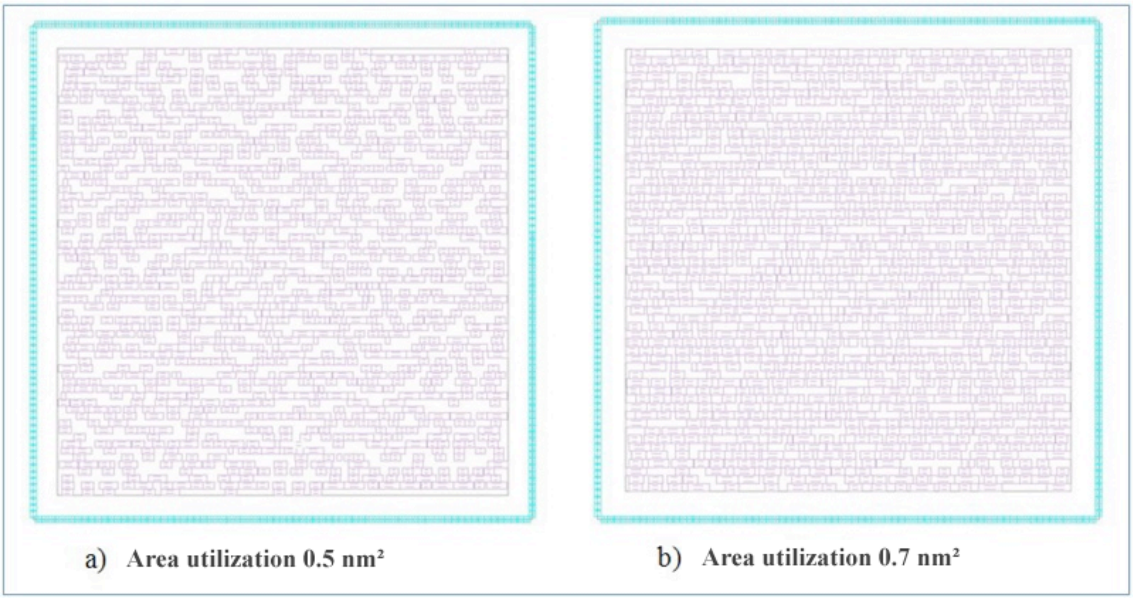

Fig. 1 Standard cell placement showing the area utilization of the same circuit

Based on redundancy, Triple Modular Redundancy (TMR) is used to mitigate single fault or error in integrated circuits. TMR is one of the most widely technique used; it has been proposed by Von Neumann (NEUMANN, 1956), which is a well know fault tolerance technique for coping with errors in integrated circuits. Normally, TMR schemes use three identical logic blocks. Each one performs the same task in tandem with the corresponding outputs being compared through Majority Voters (MVs). However, it may not be efficient to mitigate multiple faults. TMR uses majority voters to choose the correct value by selecting two out of three. The voting is usually done bit a bit. Therefore, TMR can cope sometimes with multiple faults that may affect different bits voted by distinct majority voters. However, it is important to investigate how much faults a TMR technique can tolerate. Moreover, because of that, voters can be placed among combinatorial and sequential logic blocks, where it creates barriers to the faults. However, this approach, which uses replication of identical components, is not immune to multiple faults, for example. This work uses TMR with different granularities, this means, to insert different number of majority voters in different parts of the circuit.

It may be implemented at Register Transfer Level (RTL) or netlist level. The tool can remove redundant logic judging unnecessary or change some parts of the circuit in order to optimize. Consequently, the TMR added on the RTL level can be interpreted by the synthesis tool as repeated logic, and thus removed. Otherwise, when the TMR is added after the synthesis, it is guaranteed that the circuit will be maintained. There are different levels of implementation using majority voter, it can be only in the final output of the circuit, Coarse Grain TMR (CGTMR), or between blocks, Fine Grain TMR (FGTMR), as can be seen in the figure 2 and figure 3, respectively.

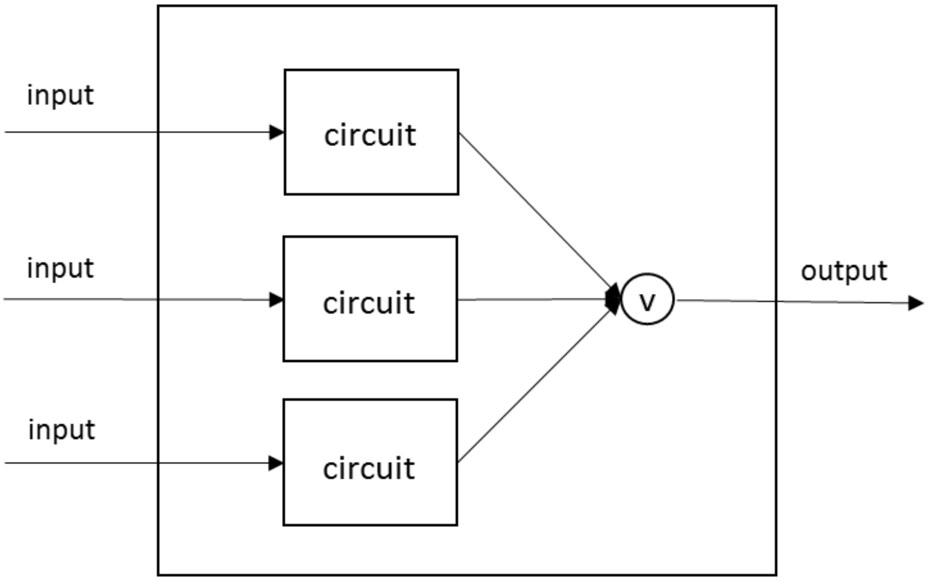

Fig. 2 Circuit protect by CGTMR

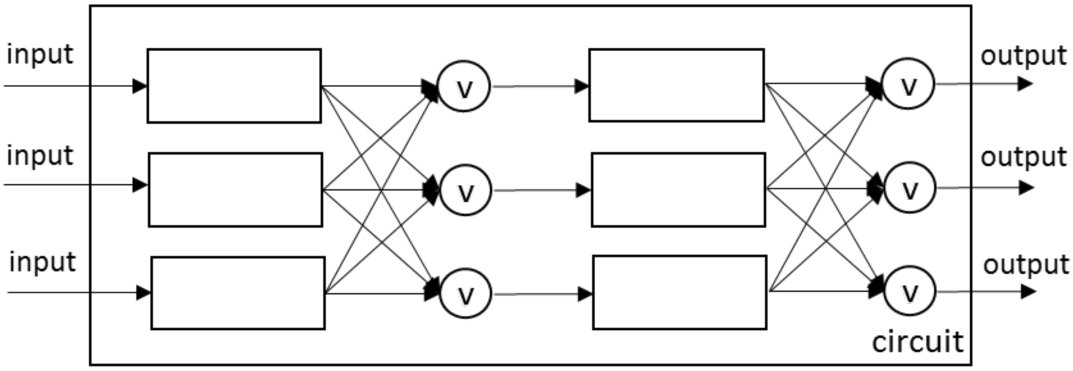

Fig. 3 Circuit protect by FGTMR

In a multiple fault scenario, TMR may not always present an acceptable level of fault tolerance (SAMUDRALA, 2004). One of the main problems is single-event-induced charge sharing, an effect that occurs when multiple faults are created, generally at physically adjacent circuit nodes by a single particle strike. They can manifest themselves as Single-Event Effects (SEEs), affecting either combinatorial or sequential nodes (OLSON, 2005).

When considering single-event-induced charge sharing, there is one primary node that receives the primary impact, and the neighboring secondary nodes that can collect part of charge deposited by the particle that struck the primary node (MASSENGILL, 2007). The number of multiple upsets will depend on many aspects such as deposited charge, distance of the transistors in the layout, state of the transistors, charge collected by each sensitive node, fan-out of each node, logic masking, electrical masking, and the latch-window masking. In addition, there is the pulse quenching effect (AHLBIN, 2009) that must also be considered.

The problem of mitigating multiple upsets cannot be solved with the use of TMR solely because multiples upsets can occur in different modules depending on the placement. That being said, a constrained placement methodology was presented in (ENTRENA, 2012). It takes into account the information about the placement of standard cells in the circuit layout and it identifies certain standard cells that must be placed far away from each other in order to reduce the error rate. The limitation of this method relies on the placement tool that does not always honor these distance constraints that have been specified (placement tools target is usually to optimize other circuit parameters and characteristics, such as area, routability and delay). In this case, TMR with different levels of granularity can be used to improve the probability of masking multiple faults in the circuit. The placement of the cells combined with an optimal TMR partition with the majority voters are very important factors to reduce the probability of errors due to multiple faults.

The use of hardware redundancy by itself is not sufficient to avoid error by multiple faults, and it is mandatory to reload constantly the system to avoid the accumulation of faults. So, an injection methodology of multiple faults has been presented. It takes into account the real position of nodes in the Application Specific Integrated Circuit (ASIC) circuit. Moreover, a study using TMR verifies when an insertion of majority voters can be used to increase the reliability of TMR to multiple faults.

The proposed methodology uses TMR hardening under various levels of granularity. Constrained placement is also used to minimize the Soft-Error Rate (SER) due to single-event-induced charge sharing. The design flow of an ASIC is commonly performed by a set of Computer Aided Design (CAD) tools that automate the synthesis of a hardware description language design into a netlist of standard cells. The Integrated Circuit (IC) implementation flow involves several steps and changes of one company to another, but it is very similar with other flow, to re-programmable device, such as a Field-Programmable Gate Array (FPGA) flow. Essentially a generic flow can be described as following in figure 4.

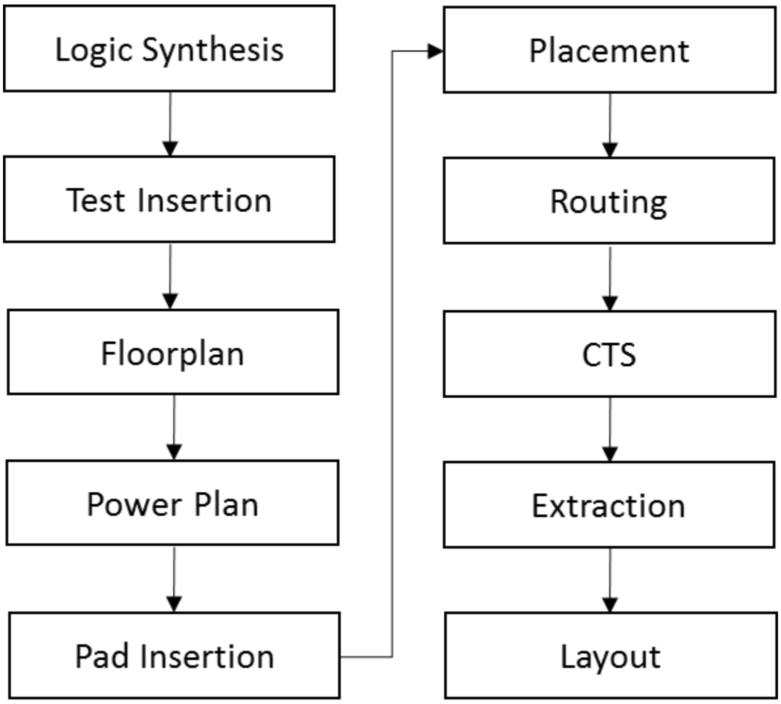

Fig. 4 Generic ASIC flow

Each one of the tasks is a transformation, which the current design goes through. We modified some of these steps in order to implement the techniques used in this work. One alternative design flow is performed to introduce the techniques implementation.

The design flow is performed once to generate a first draft layout to be analyzed by AMUSE. This tool will generate a list of SER rate and a list of the most sensitive pairs of nodes. With this information, it is possible to build a set of placement constraints and perform again the design flow. Two constraints can be setup. One is the floorplanning constraint file, which may force two or more standard cells to be placed as close as possible; the other is the area usage constraint, which defines the percentage of area usage for the standard cells available in the core surrounded by the pads. The lowest parameter is, for example, 0.5 nm, which means the highest distance between standard cells. According to the size of the circuit and the number of pads, the final area of the chip may or may not vary by changing the usage area parameter.

The goal of this work is to compare the addition of MV with the use of area utilization in order to achieve an optimal partition of the TMR logic and placement constraint methodology. The circuit must be well designed to have minimal number of voters and the best positioning of the cells. The aim is to maximize the fault tolerance without sacrificing the performance and area of the circuit.

This work investigates effects of single-event-induced charge sharing in various TMR schemes with different levels of granularities where MVs are placed in different locations in the design. The case-study circuit is an Advanced Encryption Standard (AES) algorithm (NIST, 2001) implemented in a RTL using VHDL. This analysis evaluates the SER, it uses a fault injection method (ENTRENA, 2012), and it injects millions of multiple faults in adjacent standard logic cells of the design, with the aim to analyze the robustness of the TMR versions. Finally, results show the best trade-offs between soft error rate, area and performance.

The dissertation is organized as follows:

Chapter 2 briefly introduces the problem of multiple faults and radiation effects on integrated circuits manufactured using Complementary Metal-Oxide-Semiconductor (CMOS) process, types of fault, and fault tolerance approaches to mitigate these effects in ASIC circuits.

Chapter 3 shows the proposed ASIC design flow aiming evaluating ASICs under multiple faults scenarios. The changes were made in generic flow in order to evaluate SER, as well as how is done the fault injection using the AMUSE on our case study, and the results.

Chapter 4 presents the improvements on placement constraints methodology for ASICs under multiple faults scenarios. How the methodology works and shows the results in a case study using only this mitigation technique.

Chapter 5 introduces the state of art of fault mitigation techniques based on modular redundancy. It shows the concepts of TMR and its limitation, Diverse Triple Modular Redundancy (DTMR), and n-Modular Redundancy (NMR) to cope with Charge Sharing.

Chapter 6 introduces the use of TMR in several levels of granularity to protect the circuits against multiple faults. The campaign of faults injection takes into account one main cell and a neighborhood within a given radius around this cell. The impact of a design that aims at fault tolerance for Single-Event-Induced Charge Sharing. The ASIC circuit was divided into logic blocks by functionality and it has been added majority voters gradually to decrease the number of single point of failure. Each block has different characteristics and importance on the final result. The goal is to compare several granularities of TMR using an amount of one to twelve groups of majority voters between blocks, and based on the results, we can improve the results obtaining a lower error rate. The techniques will use a fault injection with AMUSE and the results will be compared using a golden circuit. The application of the technique will be discussed and final results will be presented.

Chapter 7 presents a fault injection in the same case study but using FPGA. The conclusions of the dissertation are addressed in the Chapter 8, followed by the references and publications.

2 MULTIPLE TRANSIENT FAULTS SCENARIOS

Before introducing the developed work, this section is intended to introduce the state of the art about radiation effect and its potential effects. It will be presented the concepts about impact of radiation, which has been categorized the multiple fault scenarios.

2.1 Types of Radiation Faults

The types of faults are categorized according to its duration of time and storage in circuit. It can be such as Total Ionization Dose (TID), which are permanent, or SEE, which are considered transient effect faults.

The TID is over a long time, when the integrated circuit is exposed to radiation during a determined period of time. Normally, these effects are related with the intensity and time that the circuit was accumulating charge. The first satellite failure due TID was the Telstar. It was launched in July 10, 1962. Therefore, the first studies were presented in 1967. The TID, mostly due to electrons and protons, can result in device failure. In either case, TID can be measured in terms of the absorbed dose, which is a measure of energy absorbed by matter. Absorbed dose is quantified using either a unit called the *rad* (an acronym for radiation absorbed dose) or the International System of units (SI unit) which is the gray (*Gy*) = 1 *Gy* = 100 *rads* = 1 *J/kg*.

The trapped protons and electrons, secondary Bremsstrahlung photons (electromagnetic radiation produced by the deceleration of a charged particle when deflected by another charged particle), and solar flare protons are the main sources of TID. Some other particles from galactic cosmic ray ions are negligible in the presence of these others.

It primarily affects the oxide layers, which may trap charge or produce interface changes. In Metal-Oxide-Semiconductor (MOS) devices, trapped charges can lead to a shift in the gate threshold voltages. More generally, in semiconductors, interface states can significantly increase device leakage currents. Ultimately, TID provokes permanent functional failures of the device (LABEL, 1996) shown as degradation of the transistor devices as voltage threshold shifts and increase in leakage current.

SEE is one of main problems in space, it happen when charged particles hit the silicon transferring enough energy in order to provoke a fault in the system. SEE can have a destructive or transient effect, according to the amount of energy deposited by the charged particles and the location of the strike in the device (O'BRYAN, 1999). It can be subdivided in two main types: Single Event Transient (SET) or Single Event Upset (SEU).

SET are caused when a particle strikes a transistor, changing the outputs logic value in the combinational logic. SEU is a wrong value in a sequential cell, which can be caused by propagated SET or a particle strike directly in a sequential cell (bit-flip). The figure 5 exemplifies SET and SEU. When a pulse or bit-flip occurs due to a radiation energy particle, it can be propagated to entire circuit arriving in a temporal barrier (flip-flop) where it is stored.

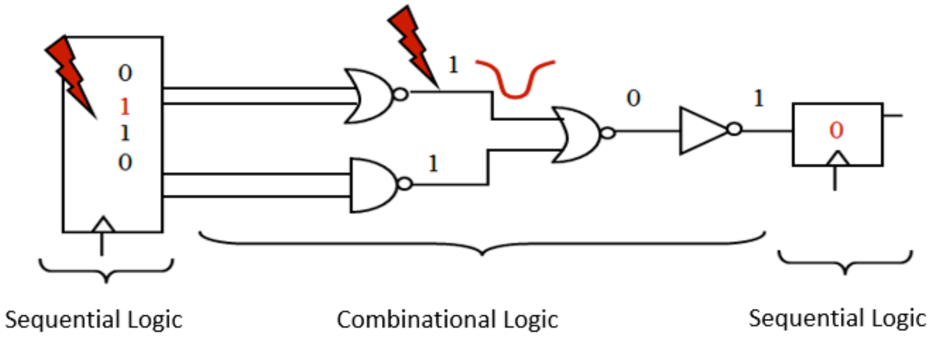

Fig. 5 Example of SEU and SET

In the space environment, spacecraft designers have to be concerned with two main causes of SEEs: cosmic rays and high-energy protons. For cosmic rays, SEEs are typically caused by its heavy ion component. These heavy ions cause a direct ionization SEE, i.e., if an ion particle hits a device that deposits sufficient charge an event such as a memory bit-flip or transient may occur. Cosmic rays may be galactic or solar in origin. (NASA, 1998)

Protons, usually trapped in the earth's radiation belts or from solar flares, may cause direct ionization SEEs in very sensitive devices. However, a proton may more typically cause a nuclear reaction near a sensitive device area, and thus, create an indirect ionization effect potentially causing an SEE (NASA, 1998).

SEE may occur due to the amount of energy of the particle that is transferred to the material, which is determined by the LET (Linear Energy Transfer). LET is a measure of the energy deposited per unit length as an ionizing particle travels through a material. The common unit is MeV*cm²/mg of material (Si for MOS devices) (BOUDENOT, 2007). The LET threshold (LETth) is the minimum LET to cause an effect at a particle fluence of the $1x10^7$ ions/cm². The fluence is the number of particles passed through cm^2.

Sensitive volume refers to the device volume affected by SEE-inducing radiation. The sensitive volume is, in general, much smaller than the actual device volume, its geometry is not easily known, but some information is gained from test cross section data.

Since both TID and SEE are from ionizing radiation, it is important to address the difference between the two with respect to design and analysis. TID is a long-term failure mechanism versus SEE, which is an instantaneous failure mechanism.

The error caused by radiation might be also destructive, known as single event hard errors, or the non-destructive, called soft errors. When talked about destructive errors, it is possible to observe that the object of this work does not compose, but comprehend the following species:

- Single Event Burnout (SEB) is a highly localized burnout of the drain-source in power MOSFETs. SEB is a destructive condition;
- Single Event Gate Rupture (SEGR) is the burnout of a gate insulator in a power MOSFET. SEGR is a destructive condition
- Single Event Latchup (SEL) is a potentially destructive condition involving parasitic circuit elements.

Regarding the soft errors or unique events, terms usually used like synonym by literature, commits a stored logic value or a signal that does not damage the device. The two more important soft error types for this work consist in SET and SEU.

2.1.1 Single-Event Transient

The well-established SET fault model is based on a single particle hitting a sensitive node in silicon, and generating a transient pulse, which changes the state of the affected node (DIEHL-NAGLE, 1984). These effects are temporary voltage glitches in combinational logic originated by the collection of charge deposited by ionizing particle strikes in the sensitive nodes of the combinational logic (BLUM, 2007).

The particle strike produces several hole-electrons under effects of the electric field. The charge generated tends to change the logic value of the struck node with a short voltage pulse, called transient pulse. However, this phenomenon can be or not captured by a memory cell or propagated throughout the circuit to the output. In other words, the SET does not always cause a failure in the system.

The figure 6 shows an example when a particle (normally protons and heavy ions cause SET) strikes in a transistor. Whether a transistor has “1” in its output and a transient current strike, the voltage pulse can be enough to switch the transistor to ground, changed the logic value to “0”.

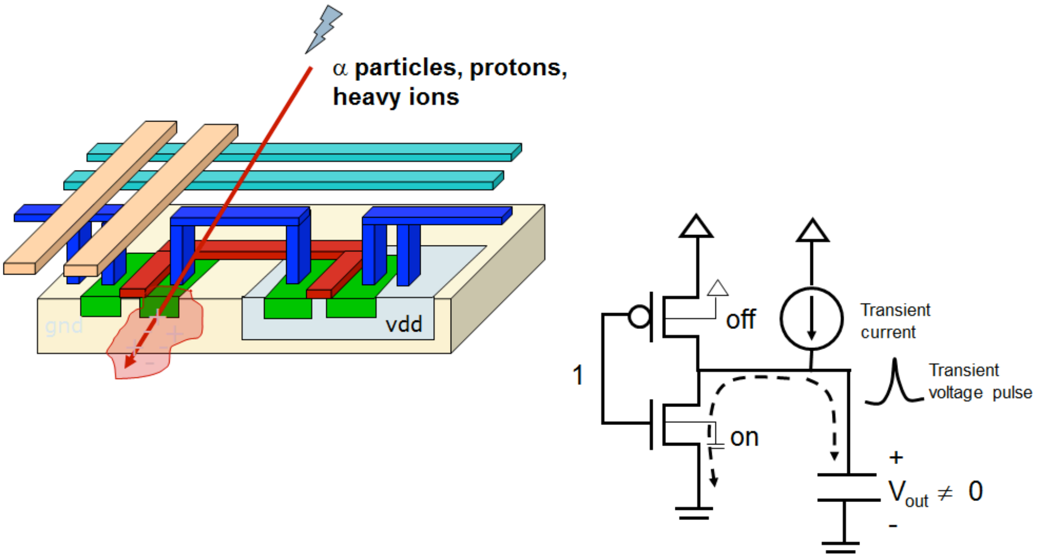

Fig. 6 Single Event Transient in transistor

As explained before, the SET may not be captured by a memory cell or propagated, not affecting the output of the system. Some effects can be observed, they can prevent a transient pulse in combinational logic from propagating to output or being stored in a memory cell. There are three main masking effects, which can prevent a transient pulse in combinational logic from propagating and being latched by a memory element: logical masking, electrical masking, and latch window masking. The figure 7 shows logic masked. It is when a particle strike in part of the combinational logic, which has not any effect in the output (ENTRENA, 2009). The output has it logic value defined by only one of the inputs, so if glitches occur in other part of circuit, the output is not changed.

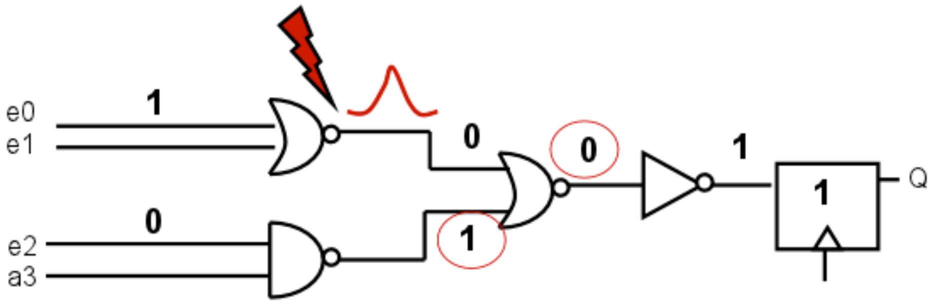

Fig. 7 Example of logical masking

Figure 8 shows the second type of masking, the electrical masking occurs when the result pulse of collision of a particle is attenuated in amplitude by whole of combinational circuit.

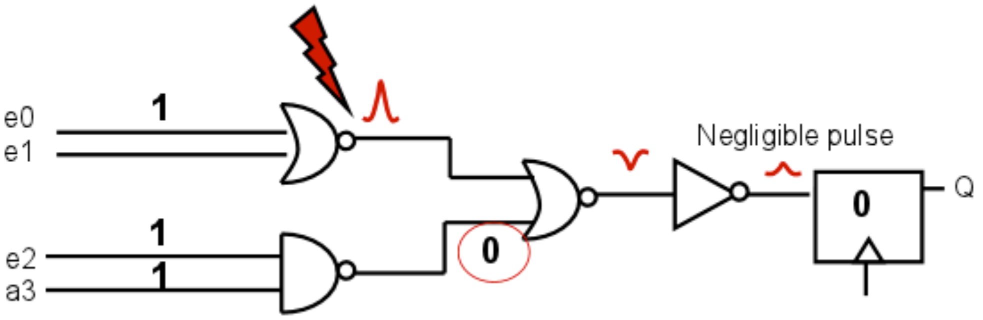

Fig. 8 Example of electrical masking

The third type is shown in Figure 9, latch window masking, and can occur when the glitch violates the times of setup and hold in a flip-flop, thus the wrong value is not stored.

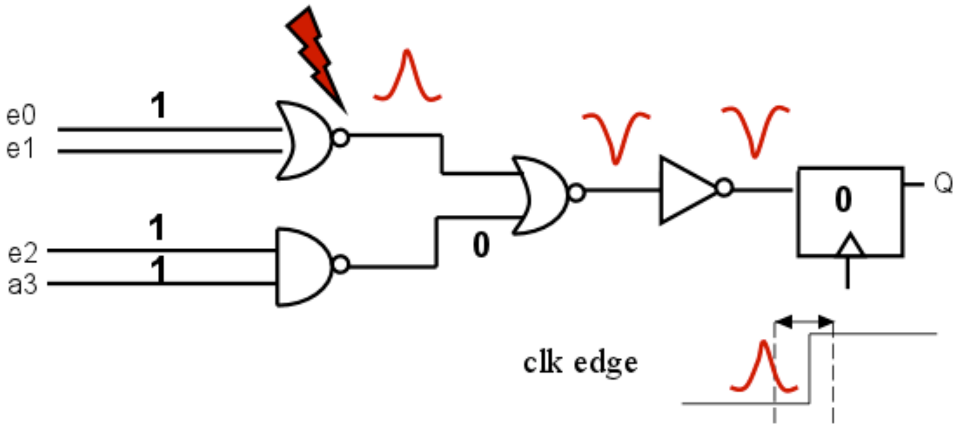

Fig. 9 Example of latch window masking

2.1.2 Single-Event Upset

Some elements are capable of storing a binary value, and posteriorly recover the bit value, if necessary. Figure 10 exemplifies this in a sequential cell, which can be a flip-flop, latch or cell memory. Generally, called of memories cells, are the most susceptible to SEU as Static Random Access Memory (SRAM), Dynamic Random Access Memory (DRAM), latches and flip-flops.

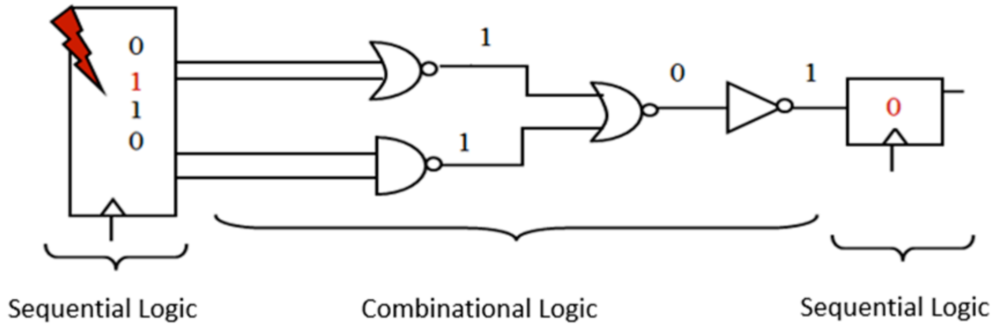

Fig. 10 Example SEU

The SEUs is the main challenge to ensure the reliability in memories cells responsible for the state change induced by incidence of heavy ions or electro-magnetic radiation in a sensitive node of a circuit. This effect becomes more worrisome if these cells are exposed to high-energy particles in space.

NASA defines SEU as "radiation-induced errors in microelectronic circuits caused when charged particles (usually from the radiation belts or from cosmic rays) lose energy by ionizing the medium through, which they pass, leaving behind a wake of electron-hole pairs." (NASA, 1967) SEUs are transient soft errors, and are non-destructive. On the other hand, it is when the charge has energy enough to change the logic value or when a SET is propagated to a memory cell being stored. In this case, the charge deposited by the particle may cause malfunction of the circuit. They normally appear as transient pulses in logic or support circuitry, or as bit-flips in memory cells or registers.

The memory elements can be DRAM, SRAM or flip-flop. When the particle strike has energy enough to change the cell value, the charge is collected in form of transient current on the transistor struck.

Faced with this problem, memory cells must be flexible to be tolerant SEU, therefore a set of cells that are derived from the standard static SRAM memory, as Heavy Ion Tolerant (HIT) in figure 11 and Dual Interlocked Storage Cell (DICE) in figure 12, for example.

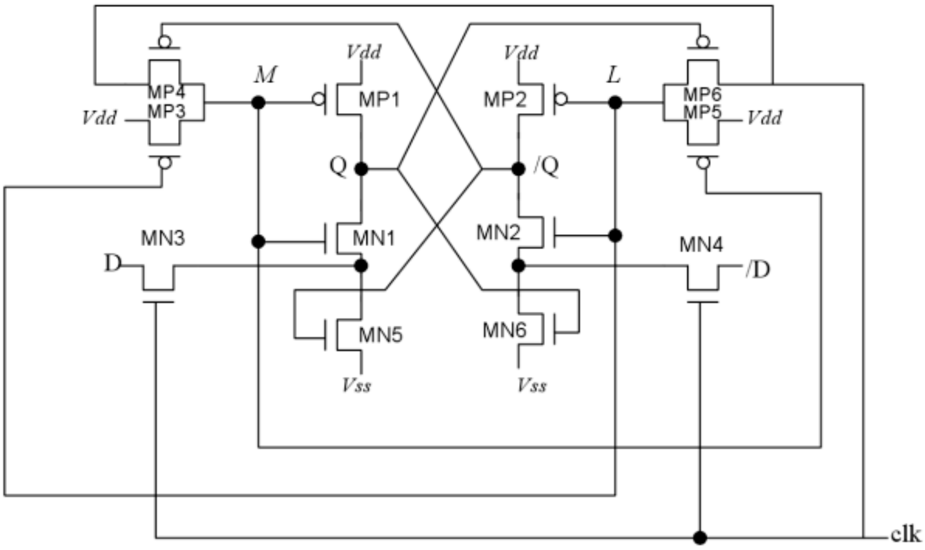

Fig. 11 Hit memory cell

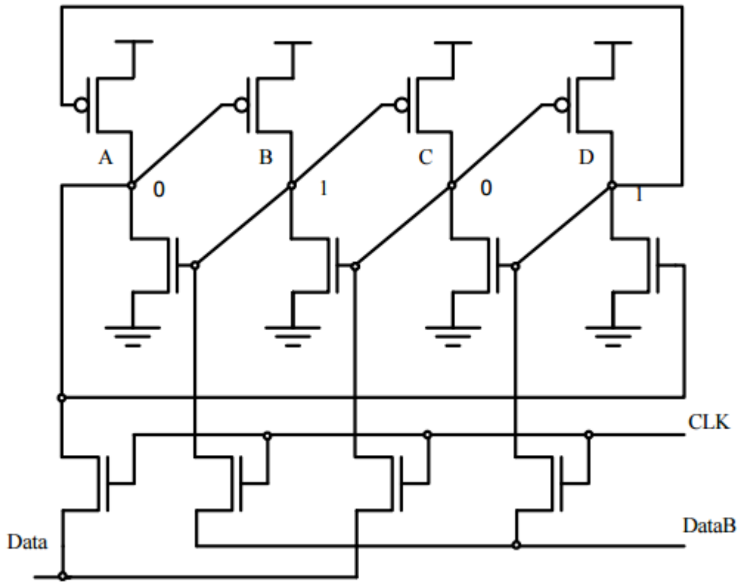

Fig. 12 DICE memory cell

These cells utilize methods to make them fault-tolerant. However, each method may use different techniques. The most widespread use feedback or duplication of information, making the data can be reversed by refeeding or by restoring the same that is storing in an isolated node, respectively.

2.2 Multiple Faults

With smaller device geometries in nanoscale technologies, it is very likely that a high energy particle strike affects several adjacent cells in a circuit resulting in Multiple Event Transients (MET) in combinational gates or Multiple Bit Upsets (MBU) in sequential elements. (RADAELLI, 2005)(GIOT, 2008) (MAESTRO, 2008)(HARADA, 2011).

Those multiple simultaneous faults are still due to a single particle hitting the silicon, in which case secondary particles can be emitted in several directions, as illustrated in figure 13 (ROSSI, 2005).

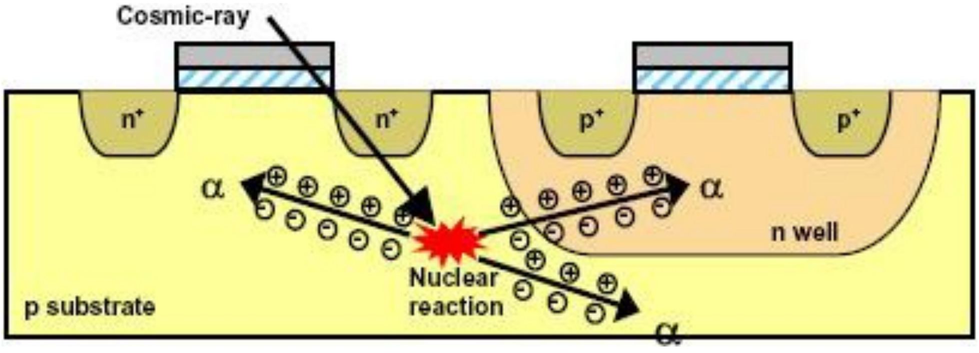

Fig. 13 One particle, multiple effects (ROSSI, 2005)

What has changed is that, since the devices are now closer to each other, those secondary particles may eventually affect two different nodes of a circuit, generating two simultaneous effects (NEUBERGER, 2003).

Moreover, after experimentally confirming that two simultaneous upsets affecting adjacent nodes can occur, Rossi (2005) has shown that the occurrence of bi-directional errors, i.e., two simultaneous complementary bit flips, will be possible, precluding the use of error detection codes designed to detect only unidirectional simultaneous errors.

One year later, Ferlet-Cavrois (2006) presented a detailed study on the charge collection mechanisms in SOI and bulk devices exposed to heavy radiation, using different technologies, from 0.25 µm to 70 nm. For bulk devices, that analysis shows that the shape and duration of transient pulses present significant variations, depending on the fabrication details, on the technology itself, and on the location in the device that was hit by the particle. Moreover, the comparison of the behavior of the same device exposed to different radiation sources has shown that some particles do not have enough LET to induce SEUs or SETs by direct ionization. However, those particles generate secondary ones, with much higher LETs, that can be emitted in all directions. Once again, the hypothesis of multiple transients generated by a single particle hit has been confirmed.

2.2.1 Single-Event-Induced Charge Sharing

Single-event-induced charge sharing that is a typical problem when talking about deepsubmicrometer technologies due to the reduced distance between devices and their small node capacitances (VELAZCO, 1996) (OLSON, 2005) (BLACK, 2005) (AMUSAN, 2007). The separation between transistors has become far smaller, then the transistors can collect different amount of charges at the same time provoking multiple SEE, which manifest itself as a combination of SET and SEU effects, depending on whether the affected nodes are combinational or sequential, respectively. As shown in figure 14, the energy can be shared by adjacent cells, hitting one or more in a certain radius.

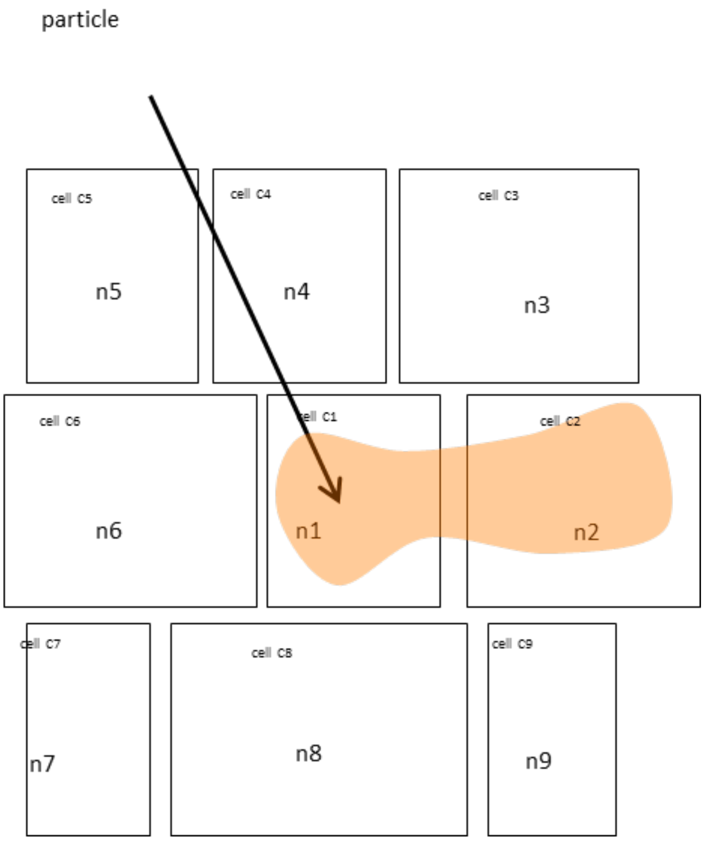

Fig. 14 Single-event-induced Charge Sharing

Charge sharing is a significant SEE issue because it can turn circuit-level hardening techniques ineffective. Multiple fault models must be taken into account to analyze error rate in high-density integrated circuits. When considering single-event induced charge sharing, it is common to have a primary impact device or active node, which is the struck hit node (i.e., hit node), and the adjacent devices or passive nodes, which are the neighboring nodes that can also collect some charge due to its proximity to the active node (MASSENGILL, 2007). For high-density integrated circuits, the active and passive nodes are not necessarily placed in the same combinational or memory cell.

Thus, it is important to analyze the effect of charge sharing not only inside a logic cell but also the impact of multiple faults in multiple logic cells in the circuit. Related works on charge sharing focus on determining the amount of collected charge in the passive nodes and to evaluate the effect of the charge sharing inside a single combinational logic gate and/or memory element (AMUSAN, 2006) (MASSENGILL, 2007) (LIU, 2009) (DODD, 2003) (MESSENGER, 1982). However, no previous work has considered charge sharing among logic cells and its effect in the functional behavior of an entire circuit.

With the advance of technology, it is expected that multiple passive nodes located in distinct logic cells may collect charge generating, simultaneously, multiple SET due to the reduced size of each logic cell. Therefore, in the current technologies, it is mandatory to characterize the sensitivity of the design to multiple faults in early stages of development.

The soft error characterization for single-event induced charge sharing can be done by fault injection considering the layout floorplanning and placement. The layout information as design placement and distances between devices are crucial data to define the possible combination of nodes affected by charge sharing at the same time. It is not realistic to inject multiple random faults, as the affected nodes must be placed together in a certain minimum distance for this phenomenon to actually occur. Many parameters must be analyzed. It is important to evaluate the number of nodes affected and the different transient pulses widths generated at each node.

For large and complex circuits, there are thousands or even millions of possible combinations of multiple upsets that can occur in a circuit and it is not feasible to analyze all combinations by simulation. Consequently, it is necessary to have a fast fault injection method that allows millions of faults to be injected in a short period of time but also taking into account the charge sharing information.

2.2.2 Pulse Quenching

Another effect, called quenching, or pulse-quenching effect has been studied. This effect occurs with the interaction in a way as to truncate a propagated voltage transient, effectively limiting the observed SET pulse widths at high LET. It is related with multi-node charge collection due to a single ion hit (YANKANG, 2013). This approach was used initially to reduce the propagated single event transient in some works (AHLBIN, 2009) (ATKINSON, 2011). However, the layout cell is a factor to reduce SEE or to increase the number of Multiple Single Event Transient (MSET) due charge sharing in transistors without no electrical relationship. Let us consider the examples, as it can be seen in figure 15. The nodes N1 and N2 are placed close enough to suffer single-event-induced charge sharing effects. For simplicity, we neglect delays and assume that all involved signals are independent and have the same probability to be 0 or 1, p = 0.5.

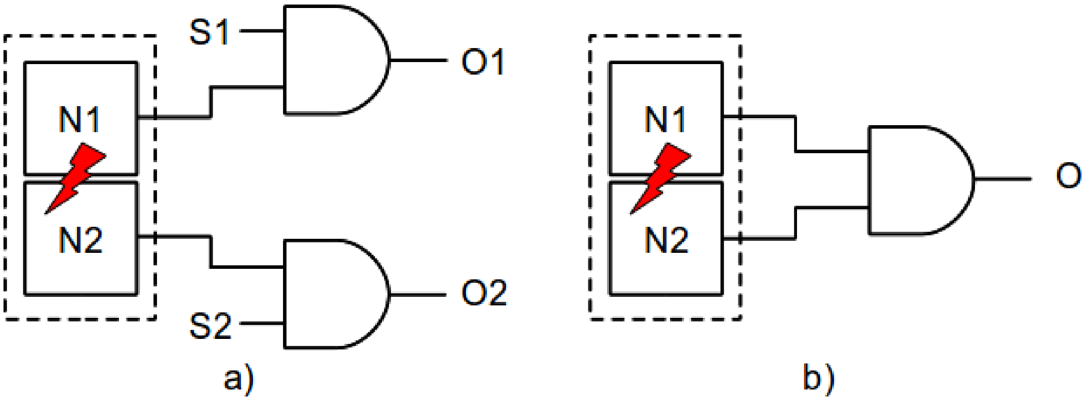

Fig. 15 Multiple fault propagation: a) Independent propagation paths b) Convergent propagation paths

The example in figure 15a illustrates the case where nodes N1 and N2 have independent propagation paths. A single fault in node N1 propagates to output O1 when the side input S1 is 1. Thus, the probability that the fault in N1 propagates to the output is p1 = 0.5. Similarly, the probability that a fault in node N2 propagates to output O2 is p2 = 0.5. The joint propagation probability that a double fault in N1 and N2 propagates to any of the outputs is p12 = p1 + p2 - p1p2 = 0.75.

As expected, the joint propagation probability is higher than the single fault propagation probability. This is an example when charge sharing has a negative effect. The example in figure 15b shows the case where the paths from N1 and N2 converge to a single output. In this case, propagation conditions are not independent. Let v(X) is the logic value at node X. The state of the circuit at a particular time is represented as v(N1)v(N2)/v(O). Using this notation, there are four possible states with the same probability:

- 00/0, 01/0, 10/0 and 11/1.

A single fault in N1 changes these states into:

- 10/0, 11/1, 00/0 and 01/0.

The output O is erroneous in two cases, namely the second and the fourth. Thus, p1 = 0.5 as in the example in figure 15a. Similarly, p2 = 0.5. In the case of a double fault in N1 and N2, the faulty states are 11/1, 10/0, 01/0 and 00/0. The output is wrong in the first and fourth cases and the joint probability is p12 = 0.5. Thus, the joint propagation probability is lower than in the example in figure 15a and is the same as the single propagation probability. This is because when the logic values at N1 and N2 are different, the errors cancel each other and produce a correct output. Note that if these states are more probable than the others, the joint propagation probability can be even lower than the single propagation probabilities. For instance, if the probabilities of each state are 0.1, 0.4, 0.4 and 0.1, respectively, then p1 = p2 =

0.5 and p12 = 0.2. This is a positive charge sharing effect. Similar results are obtained for their convergent paths with different types of gates.

These examples demonstrate that charge sharing with positive effects is possible. In a complex network, many other possible situations can contribute to reduce charge sharing effects. Although charge sharing occurs more readily between devices in the same wells (AMUSAN, 2008), for the sake of generality it will consider that any pair of adjacent cells can be upset, without making any particular assumption about the implementation that could contribute to mitigate charge sharing effects. The goal of this is to identify the pairs of cells that minimize single-event-induced charge sharing effects and use this information to guide the placement process.

The pulse quenching can be seen in figure 16, where it has two inverters that are physically adjacent in a circuit layout. In the first case, the inverters have electrical relationship. When one ions strike in an off transistor and its charge is shared, the SET pulse width is reduced due to pulse quenching effect. In order words, the logic value of the output of first invert is inverted, so the output of the second inverter is attenuated. In the second case, the output of the both inverters is inverted. This factor generates multiple SET due charge sharing.

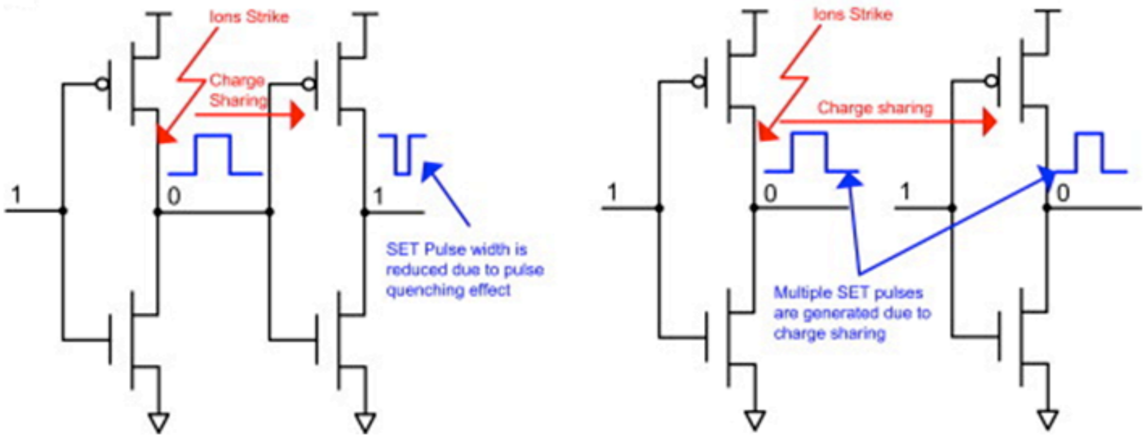
Ions Strike
Charge Sharing
1
0
1
SET Pulse width is reduced due to pulse quenching effect
Ions Strike
Charge sharing
1
0
1
0
Multiple SET pulses are generated due to charge sharing

Fig. 16 Pulse quenching effect and Multiple SET pulse (YANKANG, 2013)

Yankang at el. studied the impact of pulse quenching effect on soft error vulnerabilities in combinational circuits based on standard cells. Their simulation indicated that the soft error vulnerabilities could be reduced by 4-16% when pulse-quenching effect is introduced. They proposed an ideal optimized method to adjust the cell orientations to enhance the pulse quenching effect. This layout methodology could decrease the soft error rate when there are electrical relationships between two inverters, for example.

The two inverters are physically adjacent in a circuit layout. When these two inverters are electrically related, pulse-quenching effect could occur due to multi-node charge collection (YANKANG, 2013). This would reduce the propagated SET pulse width, which is beneficial to reduce the circuit soft errors (AHLBIN, 2009) and (ATKINSON, 2011). However, when these two inverters have no electrical relationship, MSET pulses might appear due to charge sharing. MSET pulses could shrink or enlarge the soft error vulnerabilities, depending on the circuit topology (PAGLIARINI, 2011) and (ENTRENA, 2012). For instance, when the generated MSET pulses converge at one logic cell and partially cancel each other, the SET pulse width at the primary outputs would be reduced (ENTRENA, 2012). This could lower the soft error vulnerabilities.

2.3 Multiple Faults Scenario focused on this work

In this work, we proposed a SER characterization methodology for single-event-induced charge sharing in standard-cell based designs. This methodology analyzes the effect of multiple SET at logic and system level by considering the information of charge sharing from the actual design placement using emulation.

In other words, when a particle (whether charged or not) strikes a specific spot in the circuit may or may not distribute its charge among the neighborhood of standard cells, it provoking the logic value change. In such case, the insurgent multiple faults errors may propagate them to other parts of the circuit leading to several misbehavior. Consequently, if the circuit is a mission-critical system not hardened by any faults tolerance methodology, it could lead to safety issues or even money loss.

Previous studies do not take into account the placement of cells to inject faults. A campaign of fault injection was performed randomly. In our studies, we could observe that the charge sharing happens in neighboring cells. In this scenario, this work was performed to find the best trade-off between area, performance and SER.

3 PROPOSED ASIC DESIGN FLOW AIMING EVALUATING ASICS UNDER MULTIPLE FAULTS SCENARIOS

ASIC is a design of integrated circuit made from silicon wafer for a specific application (DI FEDERICO, 2012). There are standard product or general purpose that are not as specific as ASIC, such as a logic gate or a general-purpose microcontroller, but both of which can be used in any electronic application by anybody. Examples include, chip for a satellite, chip for a car, chip for a medical IC designed to monitor a specific human biometric parameter, chip designed as an interface between memory and Central Processing Unit (CPU) and so on. The ASICs are divided in three different classes. There are:

- Custom ASIC: For this type of ASIC, the designer designs all or some of the logic cells, layout for that one chip. The designer does not used predefined gates in the design. Every part of the design is done from scratch.
- Standard Cell ASIC: The designer uses predesigned logic cells such as AND gate, NOR gate, etc. These gates are called standard cells. The advantage of standard cell ASICs is that the designers save time, money and reduce the risk by using a predesigned and pre-tested standard cell library. In addition, each standard cell can be optimized individually. The standard cell libraries are designed using the full custom methodology, but you can use these already designed libraries in the design. This design style gives a designer the same flexibility as the full custom design, but reduces the risk.
- Gate-Array ASIC: In this type of ASIC, the transistors are predefined in the silicon wafer. The predefined pattern of transistors on the gate array is called a base array and the smallest element in the base array is called a base cell. The base cell layout is same for each logic cell, only interconnects between the cells and inside the cells is customized.

When designing a chip, it is necessary to achieve the better solution to a specific application; the following constraints are taken into account:

- Performance
- Area
- Power
- Time to market

There are several kinds of design flow for different vendor (Cadence, Synopsys, Mentor and so on). Robustness is a custom ASIC but the cost and time to manufacture is larger than the others. In this work, we use the most common and cheap, which is standard logic that will be showed in this chapter.

3.1 Basic Design Flow

A simple digital flow is presented in figure 17 for a digital flow. The main goal is to understand the digital flow for sequential and combinational circuits in order to explain later in more details the modifications and additions that were performed to improve robustness to multiple faults.

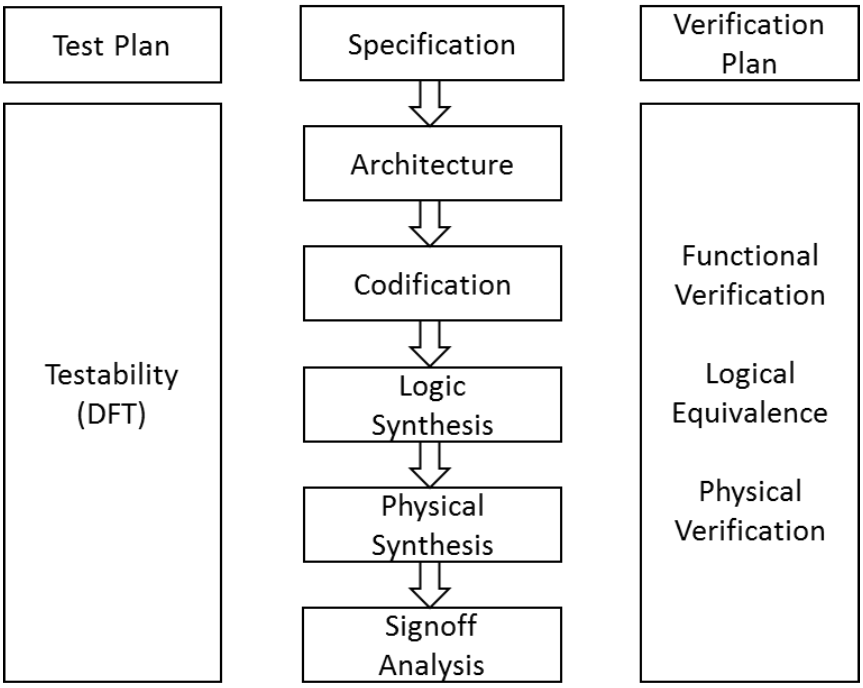

Fig. 17 Design Flow

Specification and architecture define the functionality and architecture. There are two types of specification: functional and structural. The functional specification is a formal document that describes all external interfaces and how the chip should behave. The structural specification describes all internal modules and your connections; this document directs the designer to choose the architecture and how to code it. For digital designer, how going to be Finite State Machine (FSM) or combinational circuit (Karnaugh Map).

After define the architecture, the next step is codification. Hardware languages as Verilog or VHDL are used to implement a behavior structure and achieve the all defined structural specification. The modules are defined within *process* (VHDL) or *always* (Verilog). It may be sensitive to clock or any input, sequential or combinational, respectively.

The Logical Synthesis is the procedure of translation between behavioral codes to structural netlist mapped on gates (Standard Cell Methodology). This is similar to use of boolean algebra for combinational circuits, where each operand is mapped a logical gate. Some tools can generate bitstreams for programmable logic devices (FPGAs), while others target the creation of ASICs. The tool needs RTL code and the cell library target for mapping in logic gates.

Physical Synthesis and Signoff is the layout phase. Procedures as floorplanner, placement, routing and physical verifications are done. This work focus in techniques to decrease the SER, mainly in the placement step. The tool import the outputs from logical synthesis, the design netlist and constraint file together with the technology libraries, to proceed the physical synthesis.

Testability is the insertion of extra logic that of circuit's functionality to test after manufacture; these steps are performed at transistor level schematics, and Functional Verification is done at simulation, where is created a verification environment to verify the functionality is according with specification.

The figure 18 shows the mapping of HDL in logic gates. The logical synthesis needs, at least, three inputs: Timing library (.lib), time constraint (.sdc) and HDL. It can generate how many designs you want just changing the input files.

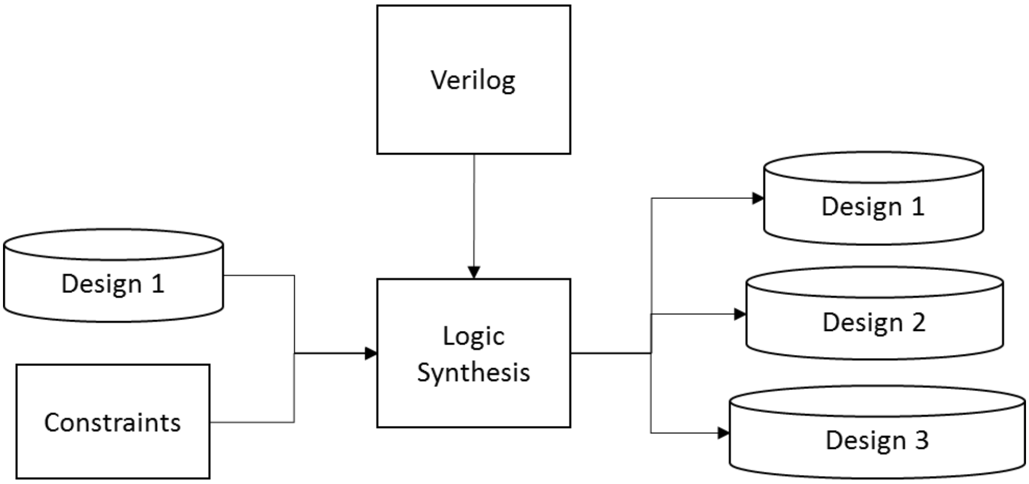

Fig. 18 Logic synthesis flow

Timing library (.lib) file is an American Standard Code for Information Interchange (ASCII) representation of the timing and power parameters associated with any cell in a particular semiconductor technology. The timing and power parameters are obtained by simulating the cells under a variety of conditions, and the data is represented in the .lib format. The .lib file contains timing models and data to calculate:

- I/O delay paths
- Timing check values
- Interconnect delays

Time constraint (.sdc) is a format used to specify the design intent, including the timing, power and area constraints for a design. Synopsys Design Constraint (SDC) is tcl based that is a guide for logical synthesis to choose the right gate size. The Library Exchange Format (LEF) is added to synthesis logic, and it is important because influence on susceptibility to single and multiple faults.

Command Types, normally are the operating conditions, wire load models, system interface, design rule constraints, timing constraints, timing exceptions, area constraints, multivoltage and power optimization constraints and logic assignments. It is used to help the tool estimate better these parameters.

Therefore, the tool reports the netlist that will be loaded in the next phase of project, physical synthesis. Backend loads the netlist in tool to generate all reports to manufacture design are described.

Physical Synthesis makes use physical layout and timing information of the target device in order to achieve the minimum area usage at the required speed. The figure 19 shows a kind of physical synthesis.

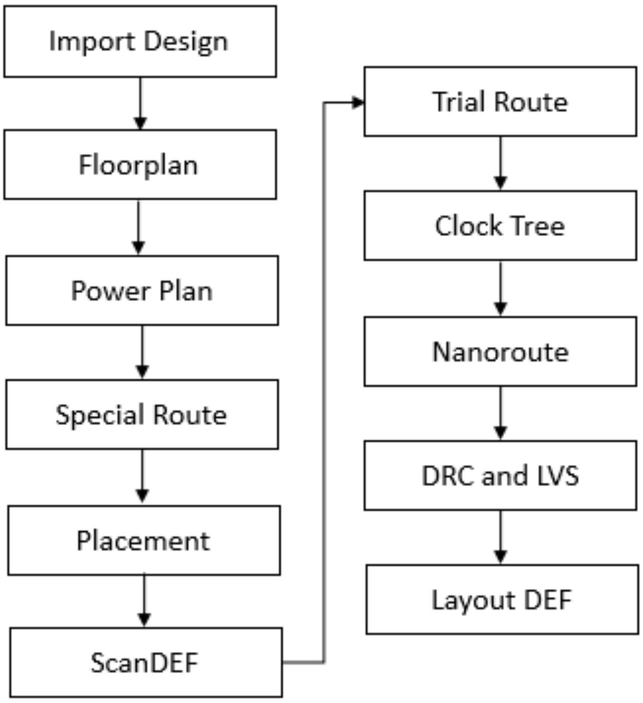

Fig. 19 Physical Synthesis Flow

Before starting the floorplan, it is necessary to load all information related of the netlist, libraries, LEFs, SDC and IO files.

The designer must create the SDC and IO files. In addition, to add all libraries. Finally, the designer needs to define power and ground nets to share out for whole the circuit.

The physical design begins with a floorplan, to specify the utilization to derive the core size of the design and estimate wiring lengths and wiring congestion. Floorplanning takes into account the macros used in the design, memory, other IP cores and their placement requirements, the routing possibilities, and also the area of the entire design. Floorplanning also decides the IO structure, aspect ratio of the design.

Before start, the floorplan is a good practice to do a time analysis (Pre-place) to avoid loading a design with some problems of the time, like a huge time slack negative, which is the difference between arrived time and capture time. In this step, you place pads, blocks and minimize cross route. In power plan, the designer creates rings and stripes for the blocks in the design, separating digital and analog blocks, which have different power structures. To create the core ring and add stripes to balance the distribution of power for whole circuit, avoid problems like ir-drop. The block pins, pad pins, pad rings and standard cell pins are connected on power and ground nets of the rings and stripes. In this phase, it is needed to verify if all power nets are connected.

The placement distributes all the cells in the design. It can specify some options like list of spare cells, Jtag cell to test and so on. All cells are pre-determined in the libraries cell, which were loaded in tool on import design. To make the flip-flops in the design controllable and observable to do test, the scan chain is inserted. The next step is the route, the tool connect all sequential and combinational logic, and can create congestion and timing violations. Routing of nets, evaluating LEF layers for the best choice. Taken into account spacing rules, routing of pins and vias, process antenna, geometry and so on.

The clock tree synthesis is the selection of cells (buffers, inverters and gate elements) to use for clock tree synthesis and run in the tool taken into account.

Concluded all this phases, another tool performs the verification of the design according to the rules from technology files, called Design Rules Checking (DRC), as well as verifying if the layout is consistent with the schematic, Layout versus Schematic (LVS).

Some analysis can be performed to verify time and power problems in different parts of the flow. As well as verification about geometry, process antenna, connectivity, density and so on. The tools take into account information on libraries and generate reports.

The final of the flow is save some files to send to foundry to be manufactured. Those files are:

- Design Exchange Format (DEF)
- Standard Parasitic Exchange Format (SPEF)
- Netlist
- SDC

The signoff analysis provides a comprehensive timing analysis and signoff verification solution that includes automated signoff Engineering Change Order (ECO), advanced modeling for precise delay calculation, power-aware, static timing analysis, accurate Signal Integrity (SI), crosstalk delay and flitch analysis, and statistical timing and leakage analysis.

3.2 Analyzing Sensitive Multiple Fault Nodes

In the literature, there are several ways to analyze the sensitivity of a device to faults. In the laboratory, it is common to use fault injection. Fault injection either at the hardware level (logical or electrical faults) or at the software level (code or data corruption) and the effects are monitored. In this work, we perform fault injection at design level in case of the ASICS case study circuit using the AMUSE platform that emulates the design in a FPGA platform to speed up the injection. The robustness measurement is analyzed in terms of SER, which means the number of injected faults able to cause an error in the output of the design.

As mentioned previously, the design flow of an ASIC is commonly performed by a set of CAD tools that automate the synthesis of a hardware description language design into a set of standard cells, performs the floorplanning, placement and routing, and finally generates a layout. In this work, we added some extra tools to the original basic CAD tool flow to analyze the neighborhood cells and to inject faults to measure the probability of error rate due to charge sharing. Figure 20 shows the flow with the added gray blocks, which is composed of the constrained placement methodology. This part was added to reduce the SER of single-event-induced charge sharing by placing the most sensitive cells far away from each other and fault injection tool to evaluate the SER.

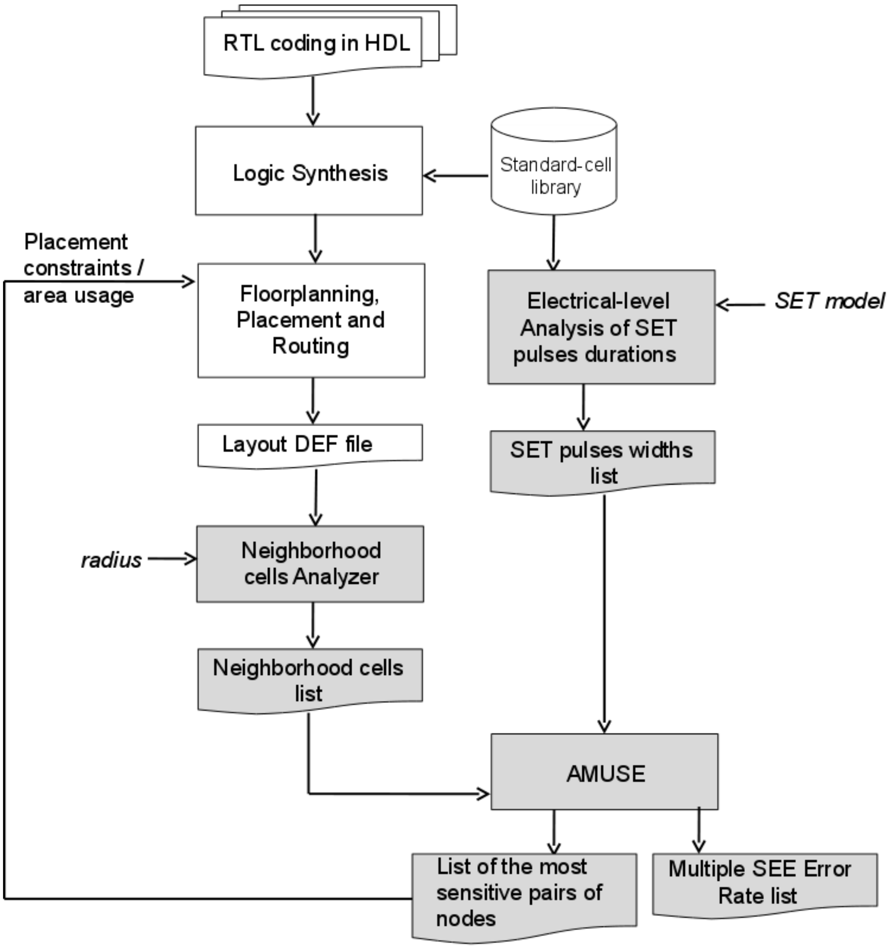
RTL coding in HDL
Logic Synthesis
Standard-cell library
Placement constraints / area usage
Floorplanning, Placement and Routing
Electrical-level Analysis of SET pulses durations
SET model
Layout DEF file
SET pulses widths list
radius
Neighborhood cells Analyzer
Neighborhood cells list
AMUSE
List of the most sensitive pairs of nodes
Multiple SEE Error Rate list

Fig. 20 The proposed methodology to reduce SER in multiple faults using placement constraints

The first layout is saved in a DEF file. The file is used as the main input to the neighboring standard-cell analyzer tool, proposed in (PAGLIARINI, 2011). This tool receives as inputs the size of the radius and the number of nodes in which the charge sharing effect is considered. At the end, the tool generates a list of neighboring nodes that is composed of the primary node and others standard-cell nodes that are struck at the same time when single-event-induced charge sharing effect occurs. Therefore, the proposed methodology uses placement information to generate a fault injection list that correlates better with the actual physical behavior. Then, based on the list of the most sensitive set of nodes identified by AMUSE, it is possible to build a set of placement constraints to be hardened in order to mitigate single-event induced charge sharing effects and perform again the design flow. On the other hand, after this analysis, the circuit is resubmitted to AMUSE, which determines the actual SER of that optimized circuit.

AMUSE is a fault-injection system that supports SET and SEU fault injection and can be used for any ASIC technology. The main advantages of AMUSE are accuracy and performance. AMUSE uses a quantized representation of time, voltage, and delays (ENTRENA, 2009), which allows to implement arbitrary ASIC delays by means of nonlinear counters. Time advances on a time quantum basis, performed by a time quantization clock. This approach covers all masking effects, including electrical masking effects, providing accuracy close to electrical simulation. On the other hand, the quantized model can be mapped into a field programmable gate array (FPGA) to boost performance. AMUSE fault-injection rates are typically in the order of 1 million faults per second, making multibillion fault-injection campaigns feasible in a short emulation time. The figure 21 shows AMUSE diagram.

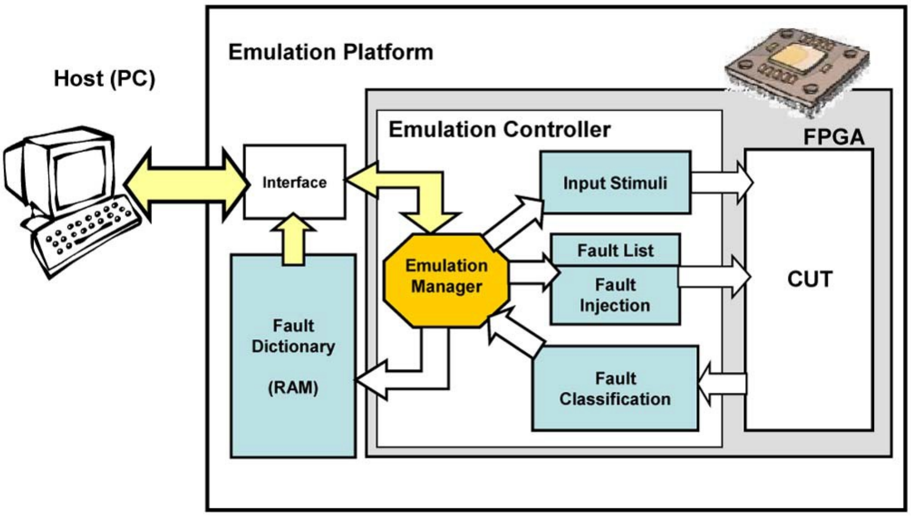

Fig. 21 AMUSE block diagram

AMUSE has been extended to support the injection of multiple SETs and SEUs. Transients of a selected pulse width can be injected at any time and simultaneously into any combination of circuit nodes. For the affected combinational nodes, the logic value is changed while the injected pulse is active. For sequential nodes, the pulse produces a bit-flip that is kept beyond the end of the pulse until the end of the current clock cycle.

AMUSE has been used to estimate the SER for every possible set of nodes in the circuit. For each SET, several thousand pulses of selected duration were injected at random instants, and their effects after several thousand clock cycles were analyzed. The fraction of faults that produced any difference at the circuit outputs was used as an estimation of the SER due to charge sharing. The collected set of SER estimations for every possible set of nodes will be referred to as the cross-SER table. The cross-SER table is a matrix where each element contains the estimated SER for single-event-induced charge sharing of node and nodes in the case they were placed close enough. The complete cross-SER table can be computed within acceptable time for small circuits, in the order of 1000 nodes. For larger circuits, a partial cross-SER table can be obtained by limiting the fault-injection campaign to the most critical nodes or, since circuits are organized hierarchically, by performing a partial analysis for each sub module of the design.

The optimal pairs of nodes can be identified by traversing the cross-SER table. In particular, we derived a set of placement constraints using the following approach. For every row, we first selected the four elements with the smaller SER. The resulting list of elements is then sorted according to the difference with respect to the average SER in its row. Finally, some elements are removed in order to ensure that no node appears in the list more than four times and that there are no redundant elements. The remaining elements constitute the set of placement constraints.

It is important to note that, for the example used in the experiments, the four elements selected for each row have a SER smaller or equal than the single-fault SER. In other words, for every node, it is possible to find at least four other nodes with positive charge sharing effects. This remarkable result demonstrates that charge sharing with positive effect can potentially be used to minimize the overall SER of a circuit due to multiple faults.

A SET can be modeled as a spurious voltage pulse on the output of a gate. The pulse may propagate across the circuit and eventually provoke malfunction. The purpose of fault injection is to inject SETs in a circuit and classify their effects. The resulting classification is an estimation of SET sensitivity for the circuit under test. In order to estimate the SER before the circuit is manufactured, fault injection is performed on the model of the circuit under test that results from the design process.

In emulation-based fault injection, the model of the circuit is downloaded into a FPGA. The voltage pulse induced by a SET is modeled at the logic level as an erroneous logic value (0 or 1) at the output of a logic gate that lasts for the duration of the pulse. Propagation of the pulse is then performed by executing the circuit in the FPGA and the fault effect is classified by comparing the result with a golden execution. Therefore, the emulation system must support fault injection at any gate and time instant, comparison between the golden and faulty execution, classification of fault effects, and external communication with the user. In Autonomous Emulation, all these functions are implemented inside the FPGA in order to improve emulation efficiency.

Propagation of a SET effect can be seen as a two-step process. First, the pulse is propagated throughout the combinational logic up to the memory elements (latches, flip-flops and memories). At this point, the SET effect can be seen as an SEU, if just one memory element or bit is affected, or as an MBU, in case several memory elements or bits are affected.

If the SET does not produce any change on the circuit state, it can be classified as having no effect. Otherwise, if the SET produces a SEU or MBU, then a second stage is needed. In the second stage, the SEU/MBU is propagated in subsequent clock cycles until the fault effect is finally classified.

3.3 Case Study

The chosen case-study circuit is a crypto core that implements the AES algorithm (NIST, 2001) and supports a range of different configurations. For the purpose of this work, a key length of 128 bits has been considered during only the encryption process. The core was submitted to synthesis using Synopsys Design Compiler and a 90nm ASIC library also provided by Synopsys (SAED90nm) (SYNOPSYS, 2004). The resulting circuit has a total of 1,191 cells, from which 11 cells are memories, 156 cells are flip-flops and the remaining ones are combinational cells. The resulting area, excluding memory blocks, which are placed separately, is summarized in Table IV.

Tab. I Details of the synthesized hardware

Total Number of Cells	Number of Flip-flops	Combinational Area	Non-Combinational Area
1,180	156	13,976.81 µm²	3,881.75 µm²

The block diagram of the AES core is illustrated in figure 22. Before any AES operation can be started, the initial user key has to be transmitted. After the user key is transferred to the component, the KEY_VALID signal must be asserted to start the key expansion. It is also required to assert the ENC/DEC signal to start encoding or decoding, respectively. Once a key is passed, DATA can be transferred by asserting the DATA_VALID signal. The result of the operation can be read from the RESULT signal once the FINISHED signal is asserted by the core. The number of clock cycles for calculating the output is 21 for a key-size of 128 bit, the only size used in our experiment.

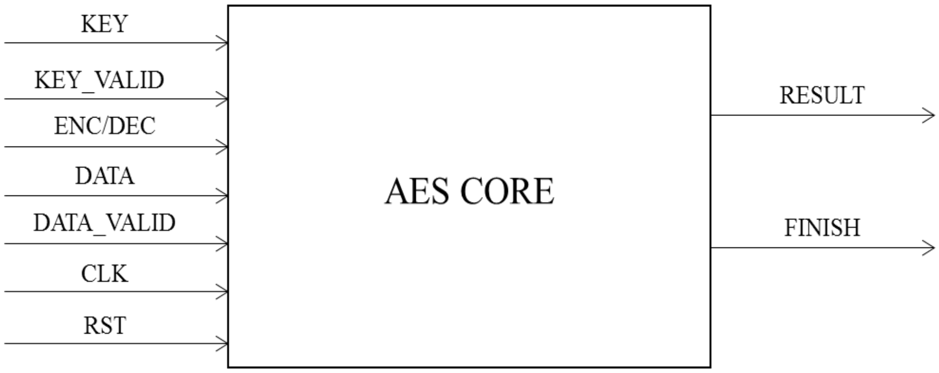

Fig. 22 AES Core block diagram

The design dimensions are approximately 150 µm x 150 µm. The average size of a single standard cell is 14 µm². In the particular technology/library pair used in our experiments, an inverter with the smallest current strength has measures of 1.92 µm x 2.88 µm, which corresponds to an area of 5.5 µm².

Two techniques are going to be applied in this case-study circuit: TMR schemes in several levels of granularities and placement constraint methodology, for ASIC and FPGA. The idea is analyze the best trade-off among area, power, performance and soft-error rate, always focusing in multiple faults due to charge sharing. On the other hand, use of FGTMR and CGTMR together with constraints to mask not only single faults but also multiple faults.

3.4 Fault injection Results

A single-event-induced charge sharing effect can be seen at a distance of up to 2 µm, when considering a 130 µm technology and the collected charge in the passive nodes can vary from few to hundreds of fC (AMUSAN, 2006). In this experiment, a radius of 2 µm is considered when double SETs are evaluated.

Several fault injection campaigns were performed to validate the proposed methodology. For each SET injection campaign, the testbench runs for 10,000 clock cycles. SET pulses were injected at every time quantum and every clock cycle, resulting in an average of 190,000 SETs into every location (single and double SET, depending on the campaign). The complete set of fault injection campaigns includes several billions of SET pulses. The entire emulation engine is implemented in a Virtex-5 FPGA prototype board and each fault injection campaign runs in few minutes thanks to the high performance provided by AMUSE.

The experiments aim to compare the SER of single and double SET, considering random and neighboring standard cells. Then, three different fault injection campaigns were performed:

- Single SET;
- Double SET into random standard cells;
- Double SET into neighboring standard cells, considering all neighboring cells within 2 µm radius.

The results of fault injection campaigns are shown in Table V. For every SET, it is reported the total number of faults injected and the percentage of errors. As the error rates for combinational and sequential nodes follow rather different trends, we present segregated results for combinational nodes only and for combinational and sequential nodes combined.

Tab. II Fault injection results

Circuit	**Upset**	**Comb. nodes only**		**Comb.+Seq. nodes**	
		#Injected faults (millions)	**Error Rate (%)**	**#Injected faults (millions)**	**Error Rate (%)**
AES	single	194.56	3.05	224.20	5.46
AES	Double (Random)	168.53	5.51	214.51	9.40
AES	Double (Neighboring)	168.53	3.86	214.51	8.80

The results of the injection campaigns showed that large overestimations might occur unless placement data is considered. On the other hand, the error rate for single-event-induced charge sharing can provoke large variations, ranging for a large increase to even a slight decrease of the error rate. The methodology proposed provides a solution to identify the most critical nodes to be hardened in order to mitigate multiple faults potentially caused by single-event-induced charge sharing effects in complex circuits. In this way, a combination of some affects nodes decrease the error rate, then we thought in a placement with more nodes together that an error could be canceled by another.

4 REDUCING SER DUE TO CHARGE SHARING BY USING PLACEMENT CONSTRAINT FOR ASICS

It is usually preferred to avoid charge sharing effects by incrementing the space between nodes or adding guardrings. An alternative solution is to avoid the propagation of multiple errors to the checking point, if possible.

Good examples of the application of this solution are memories, because they have a highly regular structure and their contents are checked one word at a time. Memories are often designed with physical separation of bits in a word, commonly referred to as interleaving. Interleaving is recommended versus simple nodal spacing because it saves area (BLACK, 2005). These types of approaches can also be used for the design of some critical cells. For instance, in (BLACK, 2005) (AMUSAN, 2009) layout mitigation techniques, such as nodal separation, interleaving, guard diodes and guard-rings are analyzed for a dual-interlocked cell latch. However, it must be noted that charge sharing effects may appear not only between transistors belonging to the same cells, but also between transistors of different but adjacent cells (AHLBIN, 2009). This problem cannot be generally solved by cell design, and requires an appropriate mitigation-driven placement approach in order to interleave critical nodes in the layout.

Another technique is custom ASIC in the layout level, which is possible manufacture cells more robustness for a determined application. This technique changes the characteristics of transistors like width and length, as well as modifying the voltage threshold in a certain range, making the transistors in the cells more tolerant to noise and radiation effects. On the other hand, it is the ideal approach but it is so expensive when include increased manufacturing and design time. The engineering costs increase exponentially, because the designer team needs to have a higher skill to develop this type of ASIC. However, the focus of this work is to use the standard cell flow, because it is less expensive and the time to develop is lower.

Then, this chapter proposes a logic cell placement optimization approach to reduce single-event-induced charge sharing effects in integrated circuits. In this approach, estimations of the error probability due to double faults are used to properly guide the placement process. In the following, we will refer to the error probability as the SER. The objective is to achieve a layout for which the SER due to double faults at adjacent cells is reduced.

A methodology able to estimate the SER of a circuit under single-event-induced charge sharing was showed in the chapter 3. This approach is able to estimate the sensitivity of each single node and each pair of nodes. The SER estimation is performed by means of the advanced emulation-based fault-injection system AMUSE (ENTRENA, 2009) (ENTRENA, 2012). Thanks to recent enhancements made to this system to support multiple bit transients (PAGLIARINI, 2011), SERs can be estimated for very large sets of multiple faults in a short time. In particular, for the example used in the experiments, we were able to estimate the SER for every possible pair of nodes with more than four-digit resolution. The experiment led to very important remarks. Not all double faults lead to a high SER. From an experimental point of view, it was noticed that charge sharing could have both negative and positive effects depending on the pair of nodes affected.

Charge sharing can have a negative effect with respect to one of the involved nodes when the SER of multiple faults is higher than that of single faults at the node. Charge sharing presents mostly negative effects, i.e., it increases the error rate. However, in some cases, charge sharing can present a positive effect when the multiple-fault effects partially cancel each other. They are less common, but they can reduce the error rate due to charge sharing with respect to that of the involved nodes. Interestingly, the experimental results show that positive charge sharing effects exist for all nodes in the example circuit. Using this information, we identified the pairs of nodes that minimize the SER due to single-event-induced charge sharing for every node in the circuit. Then, the list of the most suitable pairs of nodes is used as constraints to the placement process.

Fault injection results for a set of placements demonstrate that by placing these potential pairs of nodes together, it is possible to achieve a positive effect when charge sharing is considered, minimizing the SER due to double faults.

The goal of this work is to identify the pairs of cells that minimize single-event-induced charge sharing effects and use this information to guide the placement process.

4.1 Placement Constraints Methodology

The design flow is performed once to generate a first draft layout to be analyzed by AMUSE. The AMUSE tool will generate a list of SER estimations and a list of the most sensitive pairs of nodes. With this information, it is then possible to build a set of placement constraints and perform the design flow again. The placement constraints are based on the SER constraint file generated by AMUSE, which may force two or more standard cells to be placed as close as possible.

Once the cross-SER table has been created and the optimal pairs of nodes have been identifying, such information must be submitted to the placement engine. The placement engine used is part of the Synopsys ICC tool (SYNOPSYS, 2009). Such a tool in particular has a bounding command that can be used to create additional constraining during the placement process. For each pair of interest, a command as the following is generated:

create_bounds -name SER RULEij -effort ultra {i j}

Where i and j are cells that should be placed nearby.

Another approach for constraining the circuit is possible: identifying the nodes with negative charge sharing effects and applying rules to keep them apart from each other. However, there is no command available in the placement tool that allows for that. Thus, the chosen approach is to bring closer the node with positive charge sharing effects.

A script containing a set of constraints like the one above is created and submitted to ICC. However, such constraints are not guaranteed to be honored by any given margin since they are applied during the initial step of the placement algorithm (coarse placement). Thus, during the several refinements that are applied to the placement solution, a pair of cells might be placed more or less apart from each other.

In the first experiments reported in Section 4.2, the area of the circuit being placed, either with or without additional constraints, has been always kept the same. This is an important requirement to allow for an initial comparison that is minimally influenced by the density of cells in the circuit.

Applying the SER constraining causes the circuit placement to deviate from the (optimal) minimum wire length solution. A trade-off is established between the increase in wire length and the decrease in the SER. In our experiments, the typical wire length increase is around 8%. The execution time of the placement also increases with the number of constraints, as shown by Table VI.

Tab. III Increase in execution time due to additional constraining

# of constraints	Execution time (s)
0	13
10	14
50	15
100	14
500	15
1000	16

When the placement is finished, a layout analysis step is executed, in order to find the actual pairs that were placed in the final solution and that are sensitive to single-event-induced charge sharing effects. Such analysis is performed by the tool presented in (PAGLIARINI, 2011), which identifies all pairs of cells placed in a given sensitivity radius distance, as illustrated in figure 23. The list of pairs of nodes extracted from the exemplified placement is U89-U88, U89-U418, U89-U410 and U89-U411.

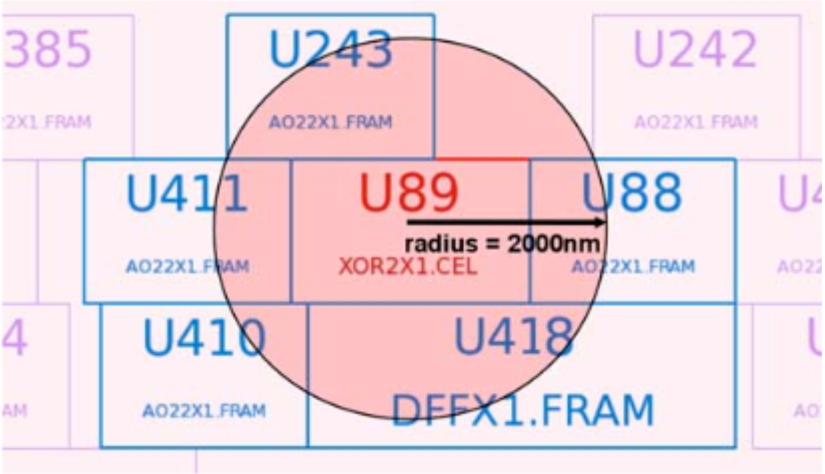

Fig. 23 Standard cell placement and selected pairs of nodes considering a certain radius distance

After this analysis, the circuit is resubmitted to AMUSE, which determines the actual SER of that optimized circuit. The following section contains the experimental results for a case-study crypto core, where a radius of 5 μm was considered for a 90-nm technology node. Based on previous studies (AMUSAN, 2006), a single-event-induced charge sharing effect can be seen at a distance of up to 2 μm when considering a 130-nm technology node, and the collected charge in the passive nodes can vary from few to hundreds of fC. For the fault-injection campaigns shown later, the considered radius is 5 μm. This radius is large enough so it could correspond to up to nine neighboring standard cells inside a cloud of collected charge. This higher radius was used to exercise the effect of particles depositing high charges in nanometer technologies. Other possible radius can also be used in the fault-injection campaign, as the methodology can be easily configured for different radius.

In order to compute the cross-SER table for the case study circuit, a set of fault injection campaigns was performed using AMUSE. A combination two by two of all of the 1180 standard cell nodes (1180x1180) was done, resulting in 1,392,400 possible pairs of nodes. For each pair, we injected 47,500 pulses of 300 ps at random instants along 10,000 clock cycles. The complete fault injection campaign included more than 66 billion faults and was executed in about 20 hours. From the campaign analysis, we then obtained approximately 1000 constraints for the case-study circuit.

To illustrate negative and positive charge sharing effects, Figure 24 shows the results for a sample node. These results correspond to a row in the cross-SER table. The SER for the selected node is 89% and it is represented in the graph by the blue line. The red line represents the estimated error rate for single-event-induced charge sharing with every other possible node in the circuit. As expected, the error rate for double fault effects (red line) is generally higher than that of single fault effects (blue line), i.e., charge sharing effects are mostly negative for the selected node.

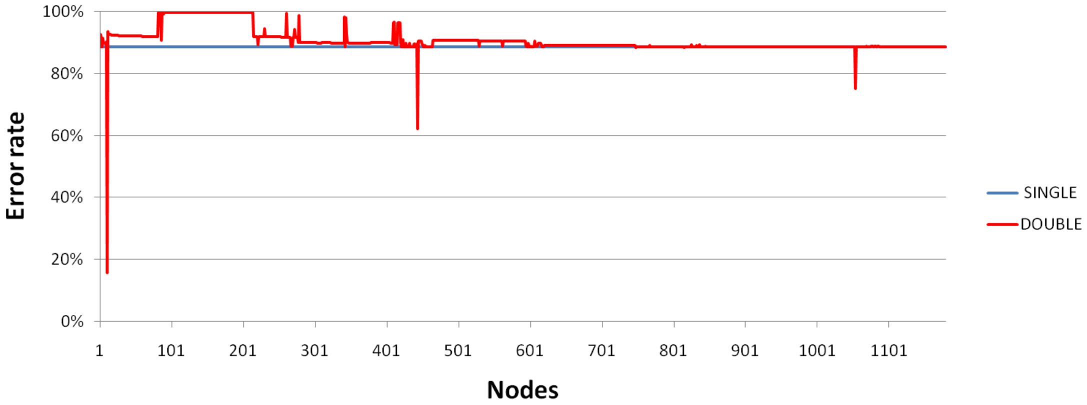

Fig. 24 Negative and positive charge sharing effects with respect to a single node

However, there are three cases for which the SER due to double fault effects is significantly reduced with respect to the SER due to single fault effects. These three cases are identified by the three notches in the graph. Thus, they are the best candidates to be placed together with the selected node. In particular, the error rate for the largest notch is 16%, which means that the SER for the selected node can be reduced by more than 5 times if charge sharing occurs between the involved nodes.

The benefits that can be obtained with the proposed approach were evaluated with three different placements of the case study circuit:

- Unconstrained placement: The placement produced by ICC tool with no additional constraints derived from the SER analysis, i.e., only those that are regular project constraints are applied.
- Constrained placement: The placement produced by ICC tool with the set of additional constraints derived from the SER analysis, as described in Sections III and IV. Note that this placement may produce suboptimal results, since the constraints are not guaranteed to be honored.

- Theoretical optimal placement: The theoretical optimal result that would be obtained if all additional constraints had been honored. This is just a theoretical result, with no real implementation, which is used just to evaluate the potential benefits of the proposed approach.

Table VII presents the percentage of errors for single and double faults for the three types of placement described above. These values represent the ratio of faults that were able to produce errors. The unconstrained placement presents an error rate 42.3% higher when double faults are injected (compared to single faults). For the theoretical optimal placement when the pairs of nodes that have a positive effect are all placed together, the error rate for double faults can be almost the same as the single fault rate.

Tab. IV SER for Single and Double faults for different placements

Circuit	SER for Single faults	SER for Double faults
Unconstrained placement		12.85%
Theoretical Optimal placement	9.03%	8.54%
Constrained placement		10.63%

However, this is theoretical because the placement tool can honor not all constraints. The constrained placement circuit is the one that has the set of constraints honored by the placement tool. For this circuit, the double fault error rate is only 17% higher than the single fault rate.

4.2 Fault Injection Results

Figures 25 to 28 show graphically a summary of all the versions. In these figures, the SER per node is presented. Figure 25 shows the SER due to single fault effects in each single node (N).

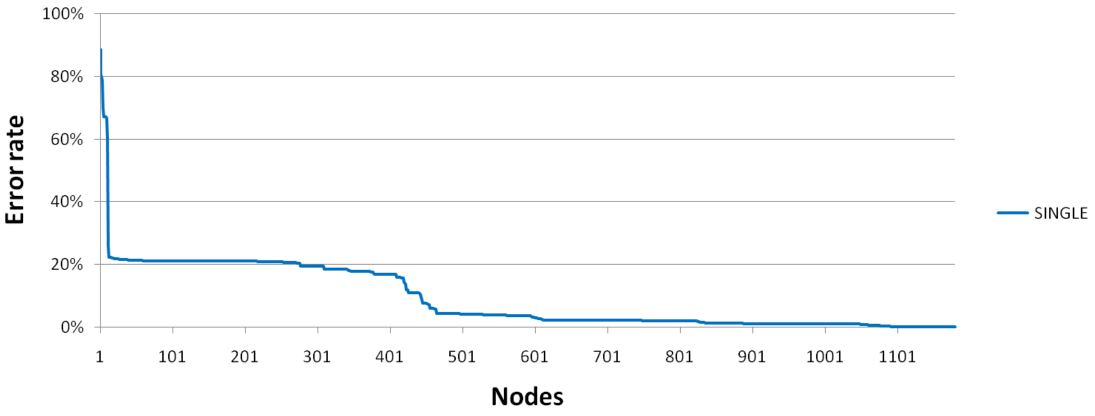

Fig. 25 SER due to single-fault effects in each single node

For the sake of clarity, the nodes are arranged in descending SER order. Figure 26 and 27 shows the SER due to double fault effects in each single node, considering the adjacent nodes in the unconstrained and constrained placements, respectively.

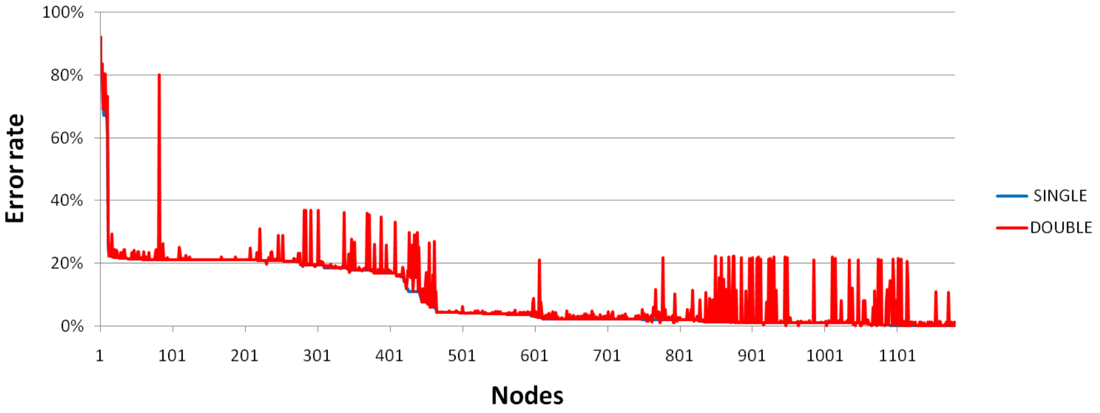

Fig. 26 SER due to double-fault effects in each single node, considering the adjacent nodes in the unconstrained placement

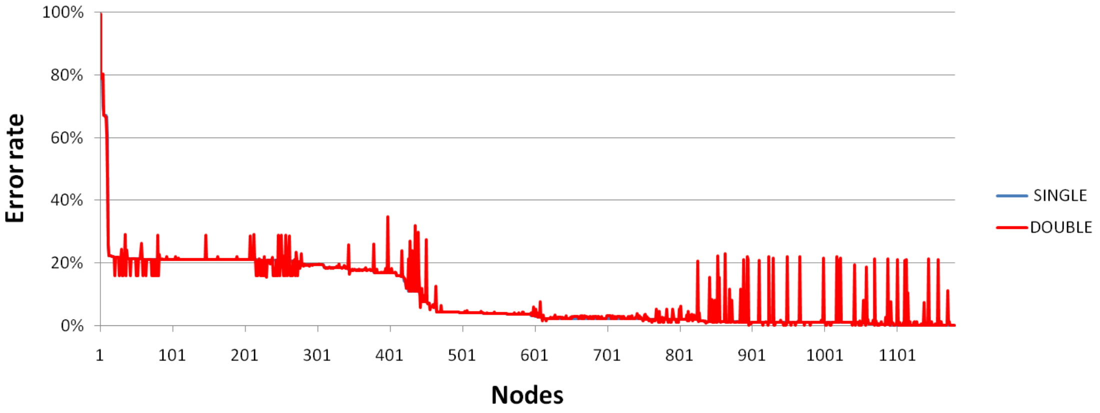

Fig. 27 SER due to double-fault effects in each single node, considering the adjacent nodes in the constrained placement

The SER due to single fault effects is also included in these figures for comparison (blue line). Finally, figure 28 shows the results for the theoretical optimal placement. In this case, four different SER estimations are presented for each node N, which correspond to the following: single fault effects (blue line); minimum of all possible pairs (red line); maximum of all possible pairs including N (green line); and average of all possible pairs including N (purple line).

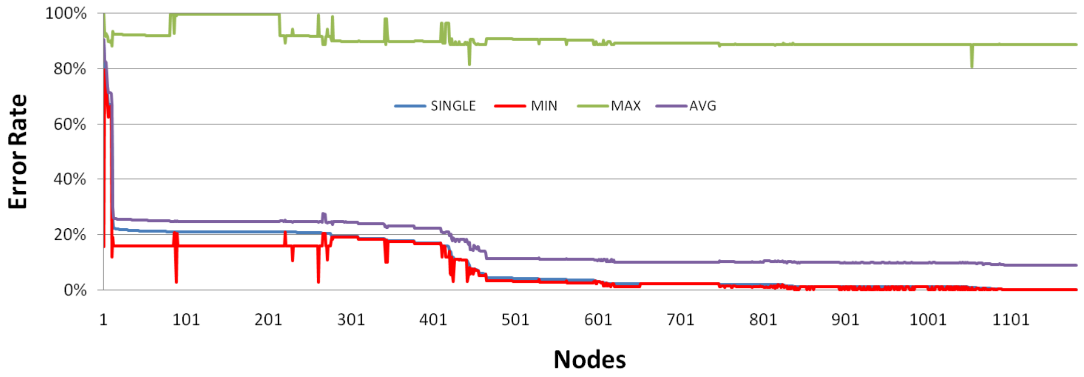

Fig. 28 Theoretical optimal placement comparison

Comparing figure 26 and 27, it can be seen that in the constrained placement many of the highest peaks have been removed and that the SER for double faults has been reduced with respect to that of single faults for many nodes. There maiming peaks correspond to constraints, which were not honored by the placement tool. This is because we used a commercial placement tool, which was not designed to give priority to the additional constraints. Figure 28 illustrates the theoretically possible results that can be obtained depending on the placement of the design. The minimum SER is always lower or equal to the SER due to single faults. However, there is a wide range between the minimum SER and the maximum SER lines, which means that a very high SER could be obtained unless this aspect is carefully considered during the placement process. Statistically, a SER close to the average line can be expected, which is well above the optimal one.

We have demonstrated that placement may have a significant impact in the soft error rate due to charge sharing and have proposed a logic cell-placement optimization approach to minimize single-event-induced charge sharing effects in integrated circuits. This approach is based on a powerful analysis of the SER for every possible pair of nodes and the generation of a set of additional constraints to be used during the placement process. With this analysis, we have also shown that charge sharing effects are not always negative. Actually, for every node in the case study circuit, we were able to identify other nodes that can reduce the SER due to double faults even below the SER due to single faults.

Integrated circuits placement tools use complex algorithms to find an optimal balance among multiple constraints, typically related to area and timing optimization. The results show that the SER due to charge sharing can be reduced by using additional mitigation-driven constraints, even though only a subset of them were honored in our experiments using a commercial placement tool.

Conclusion, the circuit has a reduced SER when we take into account the cells position, but it may not be enough if the system needs high reliability. So, the concept of redundancy is needed.

5 TECHNIQUES BASED ON MODULAR REDUNDANCY TO COPE WITH CHARGE SHARING

Generally, the amount of redundancy required to detect, mask or correct multiple faults grows very quickly with error multiplicity.

TMR is a well know fault tolerant technique for coping with errors in integrated circuits. TMR schemes use three identical logic blocks performing the same task in tandem with corresponding outputs being compared through MV. Thus TMR circuits can mask and tolerate faults that occur in one of the three logic blocks. The majority voters in the TMR perform a very important task, because they are able to mask the effects of a fault through the logic. In this way, the voters can be placed among combinatorial and sequential logic blocks creating barriers for the faults.

When charge sharing cannot be avoided using only TMR, the use of N modular redundancy (NMR) can be used, where the N correspond the numbers of redundant elements presents in the circuit. It happens that this fault technique is usually applied in different modules and not on a system as whole, where the data buses, communication, memory, analog and digital modules and processors are analyzed in the same context and at the same time.

TMR is the most common spatial redundancy technique used. There are different types of TMR with several granularities according to the number and positions of the majority voters. One can classify TMR as fine grain TMR (FGTMR) and coarse grain TMR (CGTMR), as shown in figure 29 and 30, respectively. Nevertheless, this technique allows the masking of a single fault, but it does not cope with multiple faults. Some studies have proposed the use of different granularities of TMR to improve soft error (KASTENSMIDT, 2005) (MANUZZATO, 2008) (NIKNAHAD, 2012).

Figure 29 shows a traditional TMR implementation has MVs placed only at the outputs and it is called Coarse Grain TMR (CGTMR). As mentioned before, if an error affects one of the copies, the remaining two will continue to operate properly and the majority voter can correctly mask the erroneous output of the faulty module. This actually means that CGTMR is effective to cope only with faults on a single domain, which might be practical in ground-based complex systems where it is assumed that errors upon configuration or user logic functions can occur one at a time. However, in harsh environments where SEU rates are higher, the occurrence of a high number of SEUs in a short period of time and MBUs are becoming a major concern. In this context, a coarse grain redundancy might not be sufficient to guarantee a proper reliability level, once the probability of having SEU accumulation in one module may reduce the system lifetime as a whole.

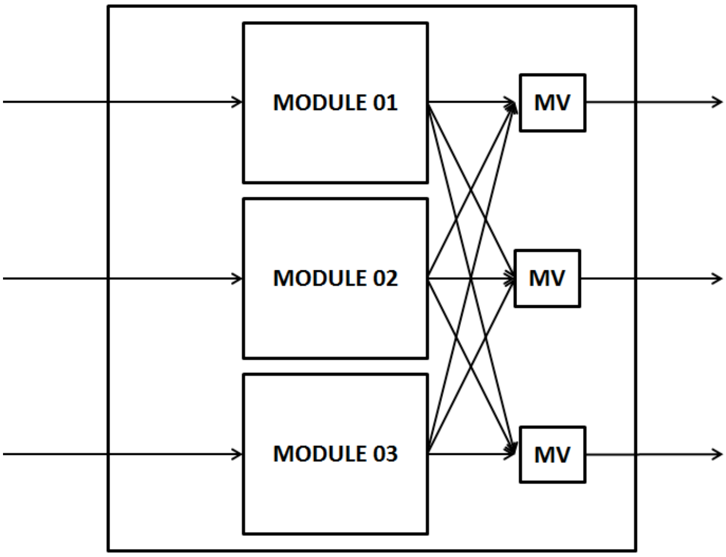

Fig. 29 Example of a TMR scheme with majority voters

As an alternative to increase the reliability of systems designs in state-of-the-art, research started to apply fault tolerance techniques to a more localized and fine level, as can be seen in figure 30. In the TMR case, this approach is usually called Fine Grain TMR (FGTMR). FGTMR in FPGA designs consists in dividing a circuit in small TMR protected blocks. As soon as just a single failure affect each small block the overall system will not be disturbed. Moreover, the probability of having multiple failures affecting two redundant modules of the same TMR system in a FGTMR scheme is lower than in a CGTMR (NIKNAHAD, 2012). Thus, an FGTMR scheme is expected to present a more tolerant scheme in the presence of a massive number of SEUs. FGTMR and a similar approach called Portioned TMR were studied in (KASTENSMIDT, 2005) (WANG, 2010) (NIKNAHAD, 2012).

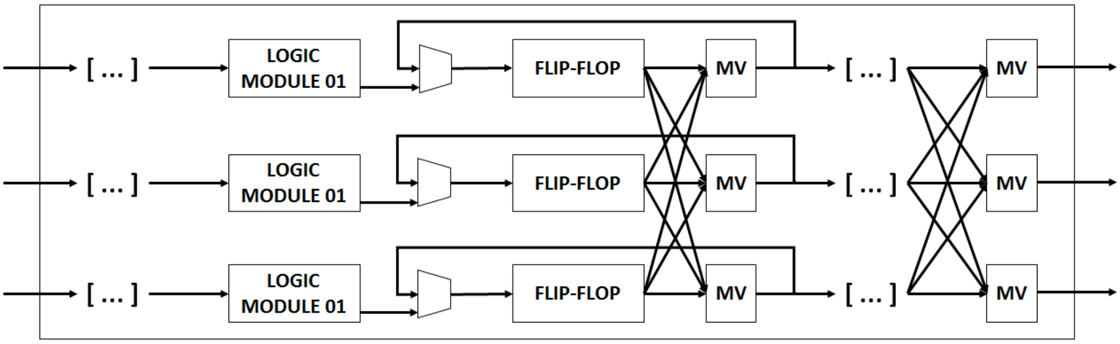

Fig. 30 Example of a XTMR scheme

5.1 Limitations of TMR

In a multiple fault scenario, TMR may not always present an acceptable level of fault tolerance (SAMUDRALA, 2004). One of the main problems is single-event-induced charge sharing, an effect that occurs when multiple faults are created generally at physically adjacent circuit nodes by a single particle strike. They can manifest themselves as SEEs affecting either combinatorial or sequential nodes (OLSON, 2005).

When considering single-event-induced charge sharing, there is one primary node that receives the primary impact and the neighboring secondary nodes that can collect part of charge deposited by the particle that struck the primary node (MASSENGILL, 2007). This problem cannot be solved with the use of TMR solely, because multiples upsets can occur in different modules depending on the placement. That being said, a constrained placement methodology was presented in (PAGLIARINI, 2011). It takes into account information about the placement of standard cells in the circuit layout and identifies certain standard cells that must be placed far away from each other in order to reduce the error rate. The limitation of this method relies on the placement tool that does not always honor all constraints that have been specified, because it takes into account other circuit parameters and characteristics, such as area, routability and delay. In this case, TMR with different levels of granularity can be used to improve the probability of masking multiple faults in the circuit. The placement of the cells combined with an optimal TMR partition with the MVs is very important factors to reduce the probability of errors due to multiple faults.

TMR can be applied in many different levels of granularity. First, one can think of a local TMR scheme, where only the memory elements are triplicated and voted. Such a scheme tolerates SEUs, but no SETs. There is also the global TMR scheme, where all the combinatorial and memory elements are triplicated and voted. In this case, it is able to cope with both SEUs and SETs. The placement of the MVs performs a very important role in the efficiency of the mitigation method. The levels of granularity can be classified by the partition of the TMR blocks that are voted out by the MVs. Each level of granularity can have a different impact on voting out multiple faults. Figure 31 illustrates a TMR with 3 partitions in the logic where majority voters are inserted. Note that according to the number of faults, the majority voter may (in the second case where the fault occurs in the same TMR level) or may not provide (in the first TMR level) the correct output. One can then change the block partition to larger or smaller blocks to try to increase the probability of multiple faults reaching different block partitions placed with distinct majority voters.

Fig. 31 TMR block partitions with multiple faults in different partitions (ALMEIDA, 2012)

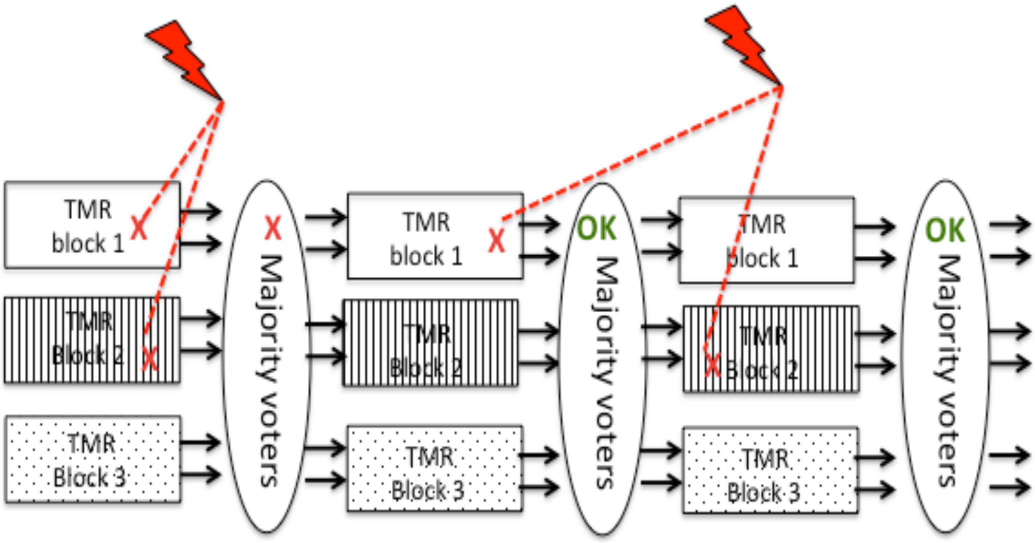

Fig. 32 TMR block partitions with multiple faults in different partitions (ALMEIDA, 2012)

When voting is not enough, changes in the circuit’s floorplan and placement might be used as well. A previous work using a constrained placement methodology (PAGLIARINI, 2011) has shown that this approach can reduce in 2.22% the error rate due to charge sharing effects. The same work has shown that the proximity of some nodes can cause regular faults along the circuit but there are also scenarios in which one fault can invalidate the other decreasing the soft error rate as a whole. In (PAGLIARINI, 2011), the authors have shown an analysis of impact of single-event-induced charge sharing in complex circuits. SER estimation is performed by means of the advanced emulation-based fault injection system AMUSE. AMUSE allows SER estimation using very large sets of multiple faults in a short time.

However, no previous studies have investigated the influence of various levels of granularities of TMR with different majority voter partitions under single-event-induced charge sharing, when placement information is taken into account.

Another technique is to use Diverse TMR (BORGES, 2010) (TAMBARA, 2013). The modules triplicated may be implemented with different architectures, such as digital, analog and so on. The idea is to create a circuit using these different architectures to implement the same feature in different copies, instead of using equal digital copies. For example: one digital copy by hardware, one analog and one digital copy by software, as can be seen in figure 32. With this process, it would be possible to reduce the probability of multiple failures affect different blocks, since each copy has a different tolerance level for faults.

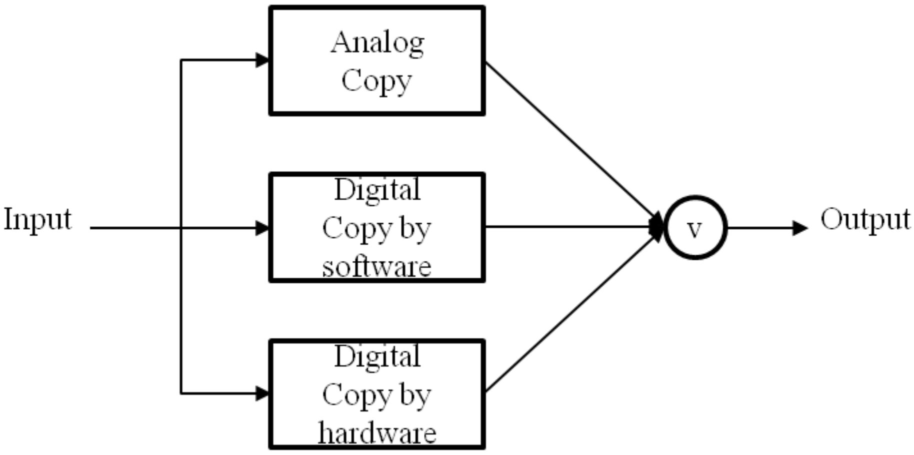

Fig. 33 Example of the DTMR scheme (BORGES, 2010)

Figure 33 shows the comparison between DTMR and a traditional TMR-MIPS scheme. The author Borges (2010) used a MIPS that is a 32 bits RISC (Reduced Instruction Set Computing) processor with a traditional TMR. The DTMR cross-section is 36.1% smaller then TMR-MIPS, which is a directly consequence of the masking factor effect.

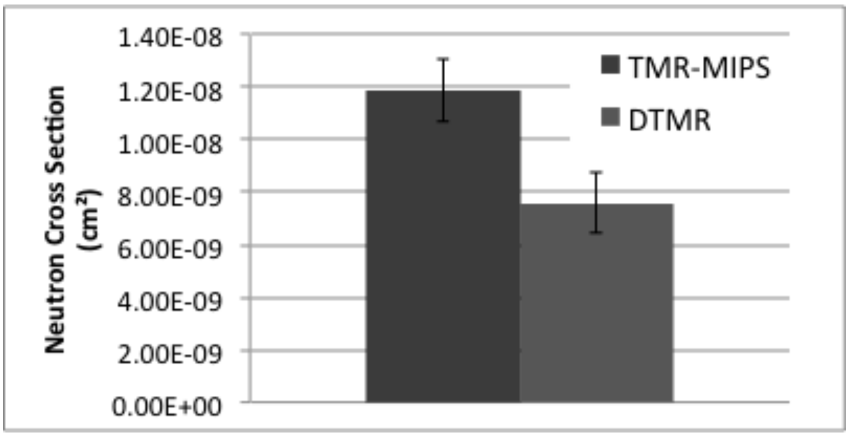

Fig. 34 Neutron cross section of DTMR scheme and Traditional TMR-MIPS

Another technique is to add redundant modules, based on the replication of n times the original module building n identical redundant modules, where outputs are merged into a voter. Usually n is an odd number higher than 3, in the case of coping with multiple faults. A use of a multiple redundancy system composed of n identical modules work in tandem and an innovative self-adaptive voter to be able to mask multiple upsets in the system was proposed (TARRILLO, 2014) to FPGA. Voter is a critical function in NMR techniques since decides the output value. Reliability of majority voters for computational structures was studied in (HAN, 2011). In (SIMEVSKI, 2012) is proposed a programmable and scalable voter for n redundancies implemented in ASIC.

In figure 34, the author proposed a scheme of NMR-based technique for tolerating multiple accumulative faults.

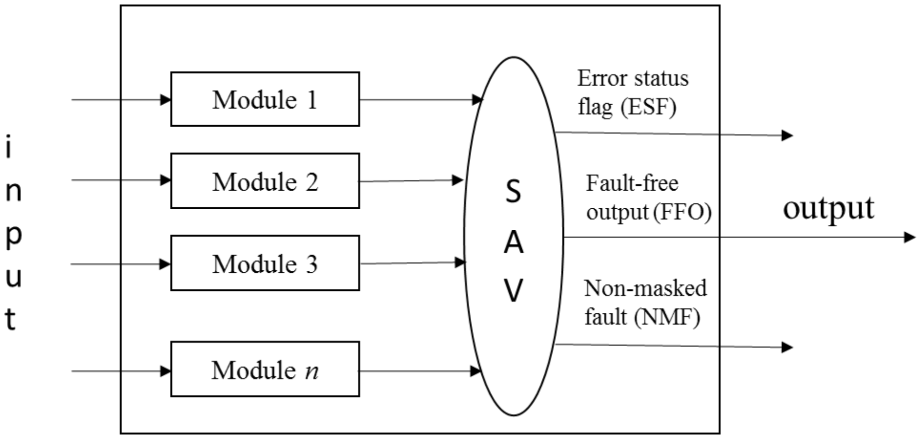

Fig. 35 NMR-based technique for avoid multiple faults (TARRILLO, 2014)

The design is done with n identical circuits that receive the same input, the output is delivered to the Self-Adapted voter (SAv). The voter generate the fault-free p-output, n Error Status Flags (ESF), and a Non-Masked Fault signal (NMF). In this scheme, the system allows for the accumulation of defective modules, while remaining at least two modules without fault.

SAv is a majority voter that considers the absolute majority as fault-free modules. The SAv and interconnections path are critical because a single fault in that structure will produce the overall system failure. However, that scenario was not considered in this work since they assumed that such elements use much less resources than the other modules, leading to a very small sensitive area.

The results were obtained using a neutron experiment test, and the figure 35 shows cross-section values for several NMR systems: *n* = 3, 4, 5, 6 and 7. As shown, cross-section falls off dramatically from *n*=3 to *n*=4 and keeps falling smoothly for greater *n*. Despite of this, the proportion of such fall off is 4.8 from n=3 to n=4, 1.27 from n=4 to n=5, 4.51 from n=5 to n=6, and 1.35 from n=6 to n=7.

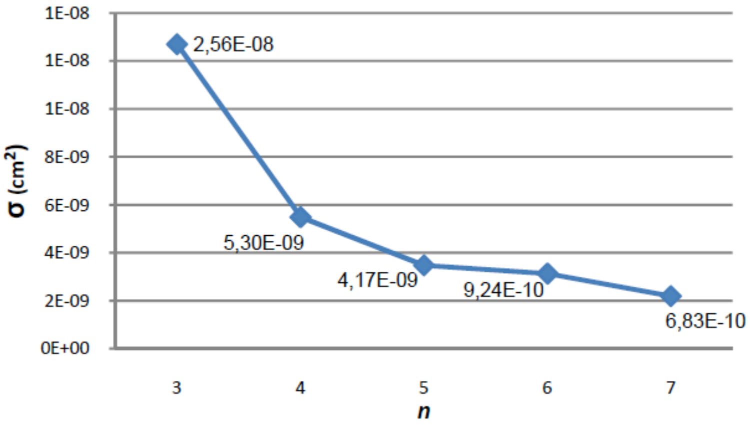

Fig. 36 Neutron Cross-section for nMR for 3...7 (TARRILLO, 2014)

According to the results, performance penalty is not affected as much as resource overhead, which was expected. On the other hand, the power consumption overhead does not increase linearly. In fact, it was shown that compared with typical TMR, in others NMR systems the increase of the power consumed (around 1.31 times) is less than the reduction in cross-section (around 37 times) for the first case, which carries a higher reliability with minimum power overhead.

Multiple fault effects cannot be considered as the simple sum of single fault effects unless the involved nodes are functionally independent (ALMEIDA, 2012). When functionally related nodes are affected by charge sharing, fault propagation can be partially reinforced or weakened. The charge sharing effects are called negative in the first case and positive in the second case. In this context, the next chapter explores different granularities of FGTMR to cope multiple faults in ASIC and FPGA.

6 EXPLORING DIFFERENT GRANULARITIES OF FINE GRAIN TMR (FGTMR) TO COPE MULTIPLE FAULTS IN ASICS

This chapter presents several granularities of FGTMR for an ASIC under multiple faults. The penalties of area, power and performance are discussed.

6.1 TMR Case Study Circuits

The case-study circuit is a cryptographic core that implements the AES algorithm with 128-bit key and data, described before in subsection 3.3. Eleven different AES TMR versions were designed and implemented by using a standard design flow from Synopsys based on standard cell libraries. They are first mapped to a 90nm ASIC library (SAED90nm) using Design Compiler (SYNOPSYS, 2012) and then floorplanned and placed. The scheme AES_v1 uses three instances of AES block with MVs placed at the output of the circuit, as shown in figure 36. The MVs vote each signal bit-to-bit.

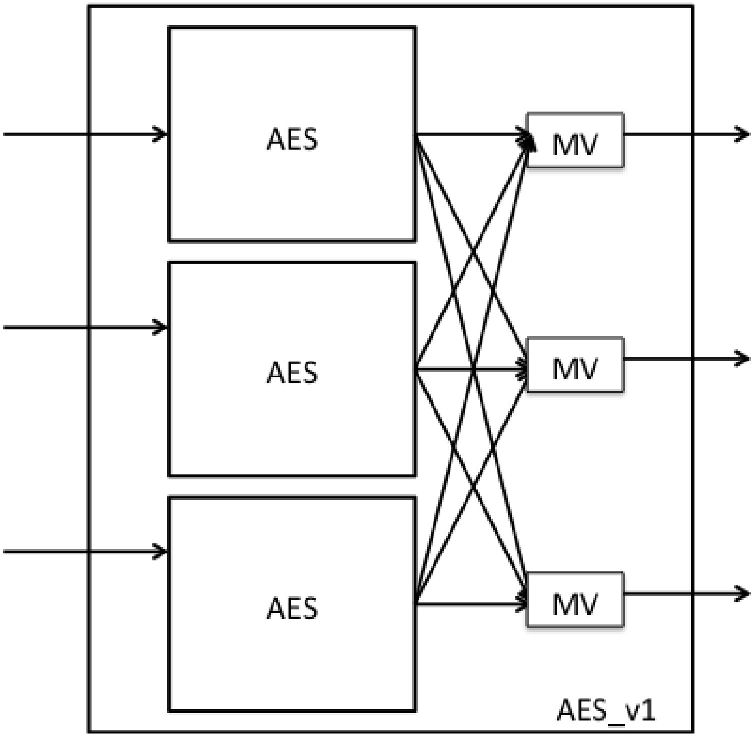

Fig. 37 AES circuits protected by TMR with large granularity (AES_v1)

The AES_v2 breaks the AES logic into small TMR logic blocks as illustrated in figure 37, and it places MVs at inputs and outputs of the TMR logic block.

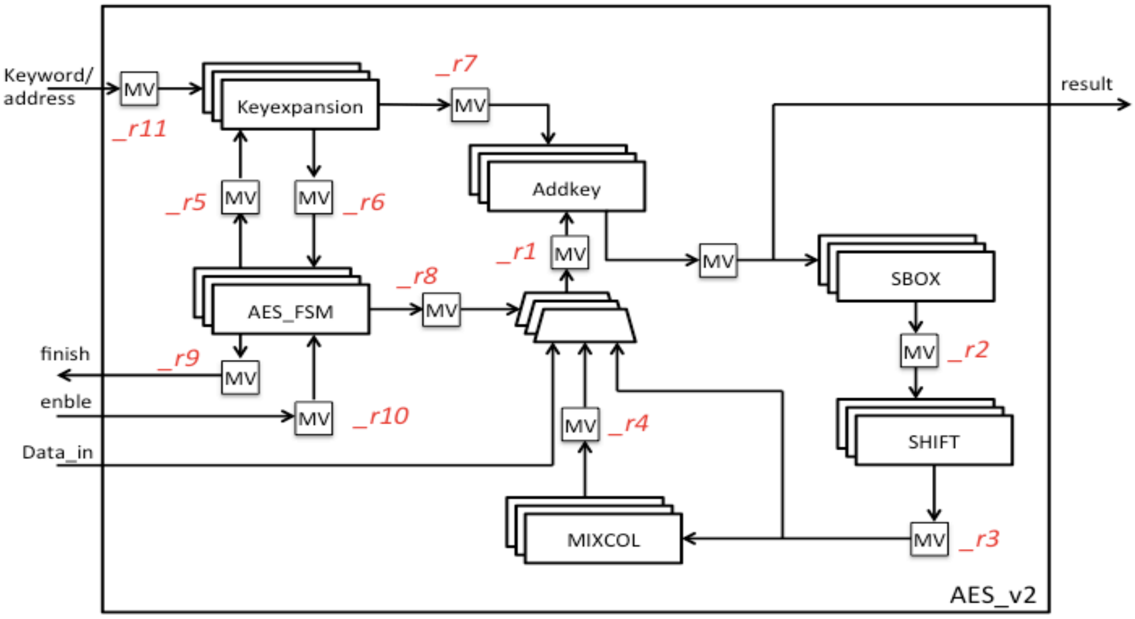

Fig. 38 AES circuits protected by TMR with small granularity (AES_v2) (ALMEIDA, 2012)

From this design, ten designs were then generated. AES_v2_v1 to AES_v2_v10 remove different MVs located at different positions in the design as described in table VIII. AES_v2_v1 removes MV _r1 for instance. AES_v2_v2 removes MV _r1 and _r2, and so on until AES_v2_10 that has only MV placed at the output of the circuit as AES_v1.

Tab. V Different AES circuits with the correspondent removed MV

Circuit	List of removed MV
AES_v2	none
AES_v2_v1	_r1
AES_v2_v2	_r1, _r2
AES_v2_v3	_r1, _r2, _r3
AES_v2_v4	_r1, _r2, _r3, _r4
AES_v2_v5	_r1, _r2, _r3, _r4, _r10
AES_v2_v6	_r1, _r2, _r3, _r4, _r7, _r10
AES_v2_v7	_r1, _r2, _r3, _r4, _r6, _r7, _r10
AES_v2_v8	_r1, _r2, _r3, _r4, _r6, _r7, _r8, _r10
AES_v2_v9	_r1, _r2, _r3, _r4, _r6, _r7, _r8, _r9, _r10
AES_v2_v10	_r1, _r2, _r3, _r4, _r5, _r6, _r7, _r8, _r9, _r10, _r11

All inputs and outputs of AES have been triplicated. Although the command *set_dont_touch* was used to avoid that the commercial tool to remove the triplicated logic, some voters' instances were still removed. So, one solution was to synthesize separately the voters and the AES core, and in other step read all netlist and merge them together again. In this way, the Synopsys tool has not removed any voters due to logic optimizations.

Results from synthesis are shown in table IX. AES_v1 has a total area of 61.293.37µm² (3.43 times larger than the non-protected version), while AES_v2 has a total area of 99.817.57 µm² (5.59 times larger than the non-protected version). In terms of performance, the AES_v1 presents a reduction in 0.03% of speed, while AES_v2 presents a reduction in 24%. As one can observe, adding MVs impacts drastically the area and performance of the design. However, at the same time improves considerably the robustness to multiple faults, as it will be seen in the next section from fault injection results are given.

Tab. VI Area and Performance of the TMR AES with different granularities and MVs

Circuit	Total number of cells	MVs (1 bit)	Combinational Area (µm²)	Non Combinational Area (µm²)	Performance (MHz)
AES_v1	4,380	12	49,648.13	11,645.24	500
AES_v2	8,289	87	88,172.23	11,645.24	380
AES_v2_v1	7,521	75	80,740.30	11,645.24	380
AES_v2_v2	6,753	63	73,308.36	11,645.24	380
AES_v2_v3	5,985	51	65,876.42	11,645.24	380
AES_v2_v4	5,217	39	58,444.49	11,645.24	380
AES_v2_v5	5,211	36	58,386.43	11,645.24	380
AES_v2_v6	4,443	24	50,954.49	11,645.24	380
AES_v2_v7	4,419	21	50,722.24	11,645.24	380
AES_v2_v8	4,407	18	50,606.12	11,645.24	420
AES_v2_v9	4,401	15	50,548.06	11,645.24	420
AES_v2_v10	4,395	12	50,489.99	11,645.24	420

Figure 38 shows part of the standard cell placement in the AES_v1 where some standard cells of the redundant block 1 (mycore1) are placed side by side with some from the redundant block 3 (mycore3). The neighboring standard-cell analyzer tool uses this placement information to extract the sensitive nodes based on a certain radius distance.

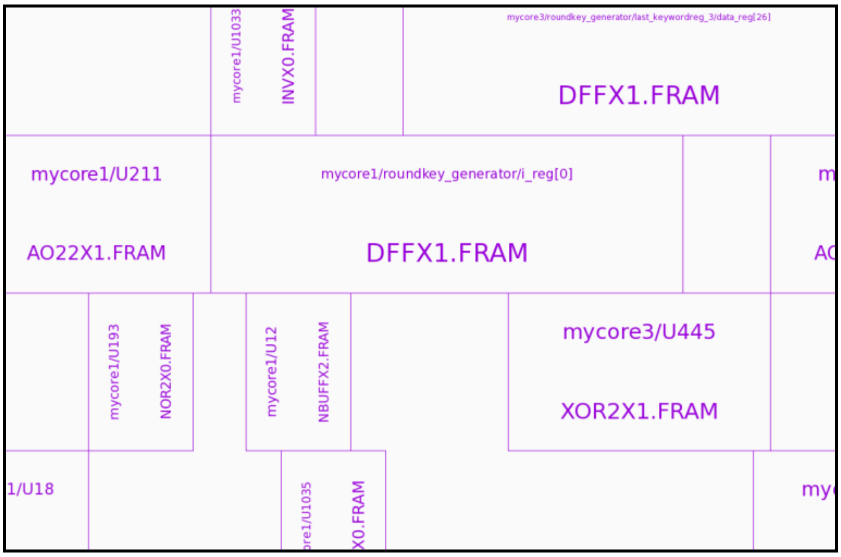

Fig. 39 Standard cell placement in AES_v1 showing the interface between redundant block 1 (mycore1) and redundant block 3 (mycore3) (ALMEIDA, 2012)

6.2 Fault Injection Results

The AMUSE tool has been used to estimate the SER in all possible combinations of nodes in the different TMR versions of the AES circuit. We have set the same clock frequency (380 MHz) in all AES fault injection experiments, although some versions of AES run in a higher clock frequency in order to evaluate them in the same conditions.

In each test case, several million pulses were injected at random instants and their effects after 10.000 clock cycles were analyzed. Following the approach described in (ENTRENA, 2012), a double exponential current pulse model is used taking into account the input logic values, the node type and strength, and the fan-out of the gate. So, the duration of the voltage pulses are determined to generate a bit-flip. The proposed methodology can be used in any radius size. In this work, a radius of 5 µm was considered. Reduced radiuses were also analyzed. However, in those cases and for that specific 90nm process technology the amount of nodes stroke would be only up to three. Consequently, because we would like to investigate groups of struck nodes up to 5, we chose the 5 µm.

First, the TMR versions with two different levels of granularity were investigated: AES_v1 and AES_v2. Table X shows the percentage of errors for single and double faults, and no error due to single faults were observed in any of the two circuits. The logic optimizations by the Synopsys mapping were performed but no voters were removed because the voters were synthesized separately as said before. AES_v1 has presented errors under double faults compared to AES_v2. This is due to the possibility of the placement tool to put together the standard cells of the same redundant block, in the case of AES_v2.

However, the large granularity presented in AES_v1 has a very important drawback when considering fault accumulation, because in this case the majority voter is placed only at the very output of the circuit. In order to improve the results of the AES_v2, modifications in the area utilization factor and in the placement constraints must be used.

Tab. VII Error Rate for single and double faults in standard and TMR designs with large and small granularities

Circuit	Upset	Combinational nodes only		Combinational + Sequential nodes	
		#Injected faults (millions)	Error Rate (%)	#Injected faults (millions)	Error Rate (%)
AES	single	-	-	224	9.50
AES	double	-	-	224	11.06
AES_v1	single	814	0.00	912	0.00
AES_v1	double	922	0.11	975	0.06
AES_v2	single	332	0.00	353	0.00
AES_v2	double	385	0.00	417	0.00

Single to multiple upsets have been injected in the eleven versions of the AES design using different numbers of voters (MV). Table XI show error rate for each type of upset (single. double. triple and multiple 4, 5 and 6) when only combinatorial (Comb.) nodes are struck by injected faults and when all nodes (combinatorial and sequential nodes) are struck. One can see that by increasing the number of voters it reduces the soft error rate, especially for multiple upsets. There are some voters that help more than other to reduce the software error rate or to increase the error due to multiple upsets that can overcome the TMR. Results show clearly that the version AES_v2, which presents the highest number of MVs and consequently the largest area, has the most reduced SER. If MVs are removed, a tradeoff can be analyzed in terms of SER and area.

Tab. VIII Error Rate for Single, Double, Triple and Multiple 4. 5 and 6 faults in TMR designs with Different granularities under millions of faults for each type of upset

Circuit	Upset	Combinational nodes only SER (%)	All nodes SER (%)
AES_v1	single	0.00	0.00
AES_v1	double	0.11	0.06

AES_v1	triple	0.76	0.88
AES_v1	Multiple 4	0.98	1.01
AES_v1	Multiple 5	1.07	2.12
AES_v2	single	0.00	0.00
AES_v2	double	0.00	0.00
AES_v2	triple	0.00	0.01
AES_v2	Multiple 4	0.01	0.03
AES_v2	Multiple 5	0.01	0.03
AES_v2_v1	single	0.00	0.00
AES_v2_v1	double	0.00	0.00
AES_v2_v1	triple	0.00	0.01
AES_v2_v1	Multiple 4	0.01	0.03
AES_v2_v1	Multiple 5	0.01	0.07
AES_v2_v2	single	0.00	0.00
AES_v2_v2	double	0.02	0.02
AES_v2_v2	triple	0.07	0.06
AES_v2_v2	Multiple 4	0.18	0.14
AES_v2_v2	Multiple 5	0.18	0.14
AES_v2_v3	single	0.00	0.00
AES_v2_v3	double	0.03	0.03
AES_v2_v3	triple	0.06	0.08
AES_v2_v3	Multiple 4	0.08	0.14
AES_v2_v3	Multiple 5	0.08	0.27
AES_v2_v4	single	0.00	0.00
AES_v2_v4	double	0.03	0.07
AES_v2_v4	triple	0.07	0.18
AES_v2_v4	Multiple 4	0.13	0.46
AES_v2_v4	Multiple 5	0.16	1.19
AES_v2_v5	single	0.00	0.00
AES_v2_v5	double	0.05	0.08
AES_v2_v5	triple	0.12	0.16
AES_v2_v5	Multiple 4	0.24	0.30
AES_v2_v5	Multiple 5	0.25	0.41
AES_v2_v6	single	0.00	0.00

AES_v2_v6	double	0.05	0.11
AES_v2_v6	triple	0.11	0.27
AES_v2_v6	Multiple 4	0.19	0.48
AES_v2_v6	Multiple 5	0.32	0.61
AES_v2_v7	single	0.00	0.00
AES_v2_v7	double	0.07	0.23
AES_v2_v7	triple	0.16	0.47
AES_v2_v7	Multiple 4	0.29	0.53
AES_v2_v7	Multiple 5	0.34	0.66
AES_v2_v8	single	0.00	0.00
AES_v2_v8	double	0.08	0.20
AES_v2_v8	triple	0.18	0.42
AES_v2_v8	Multiple 4	0.30	0.67
AES_v2_v8	Multiple 5	0.34	0.88
AES_v2_v9	single	0.00	0.00
AES_v2_v9	double	0.04	0.14
AES_v2_v9	triple	0.11	0.32
AES_v2_v9	Multiple 4	0.18	0.60
AES_v2_v9	Multiple 5	0.25	0.96
AES_v2_v10	single	0.00	0.00
AES_v2_v10	double	0.05	0.11
AES_v2_v10	triple	0.11	0.27
AES_v2_v10	Multiple 4	0.17	0.44
AES_v2_v10	Multiple 5	0.20	0.50

Figures 39 to 43 show the plotted results of error rate for double, triple, four and five multiple faults, respectively. Note that the results present the same tendency for double and triple faults, as well for larger number for multiple faults (4 and 5), increasing with the number the nodes that are taken into account. In the case of double faults, versions AES_v2 and AES_v2_v1 present zero error rate. The worst case is the AES_v2_v7. Consequently, the best trade-off in terms of error rate, area and performance would be AES_v2_v1. In the cases of triple, 4 and 5 multiple faults, versions AES_v2 and AES_v2_v1 present the lowest error rate. The worst case is the AES_v1. The best trade-off in terms of error rate, area and performance include AES_v2_v1, AES_v2_v2, but also AES_v2_v3 that presents a low error rate with a great area reduction.

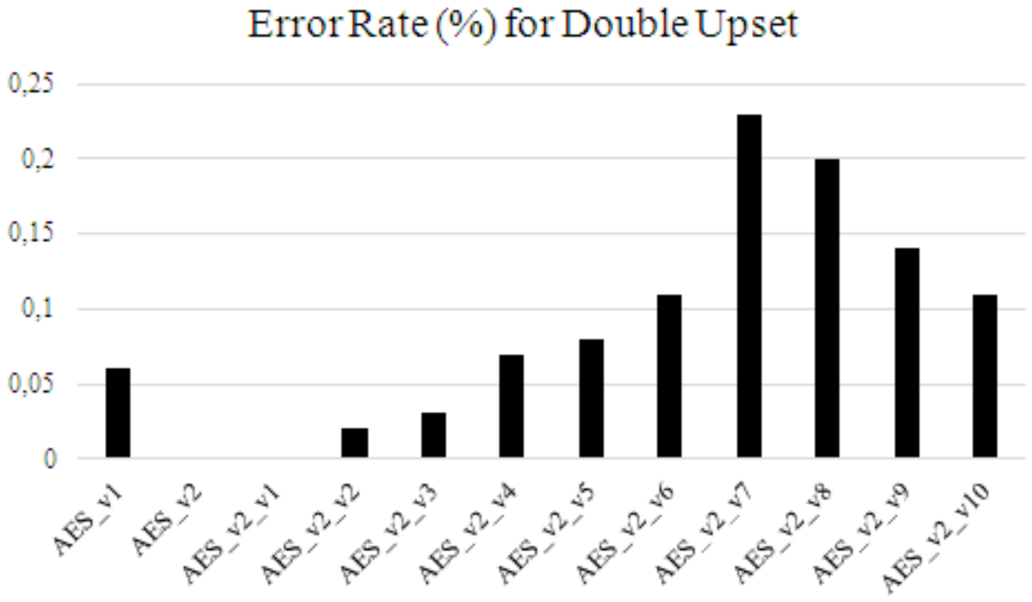

Fig. 40 Error rate for double faults in TMR designs with different granularities

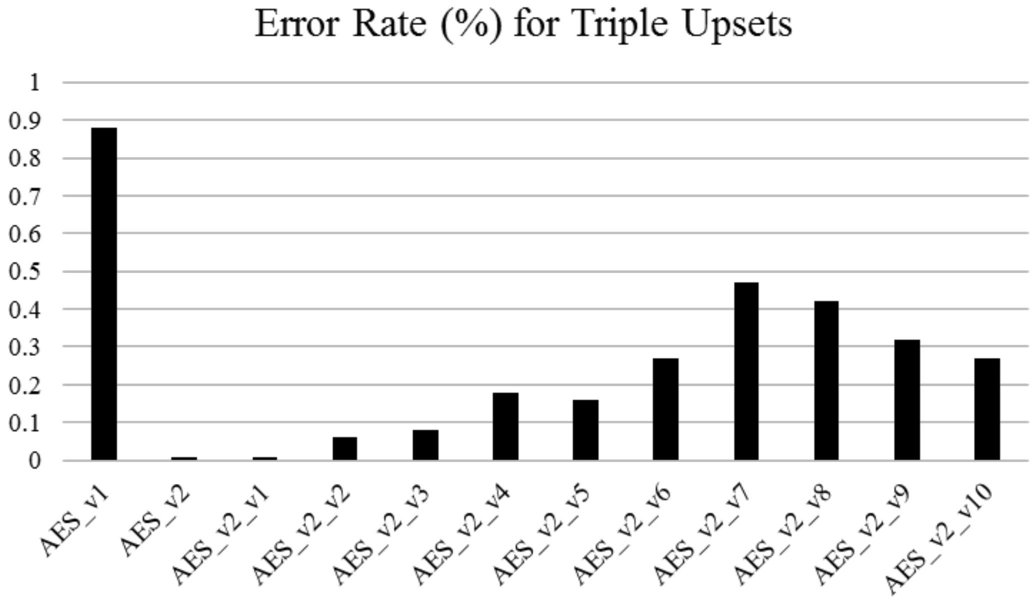

Fig. 41 Error rate for triple faults in TMR designs with different granularities

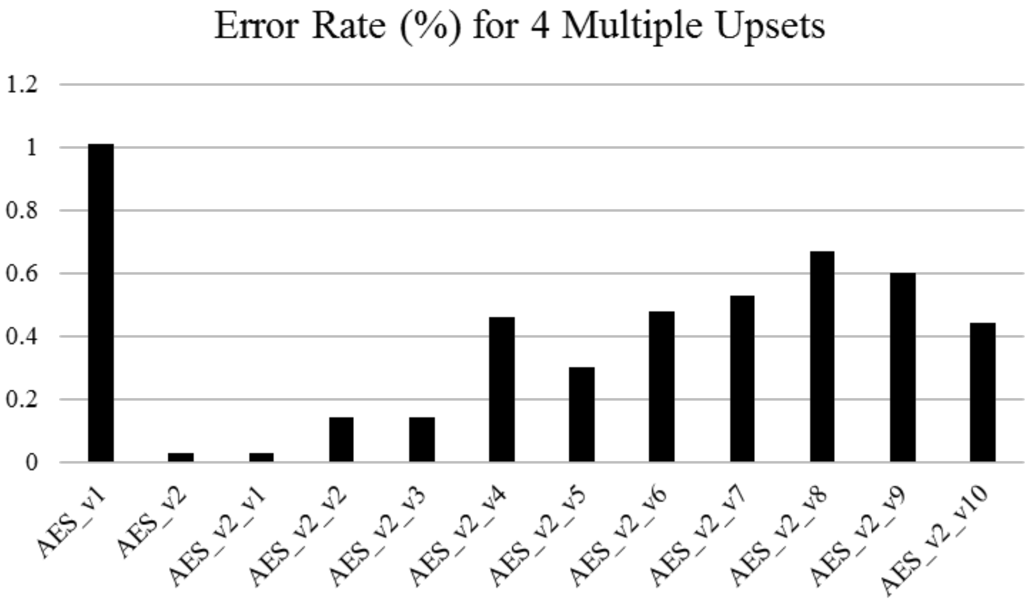

Fig. 42 Error rate for 4 multiple faults in TMR designs with different granularities

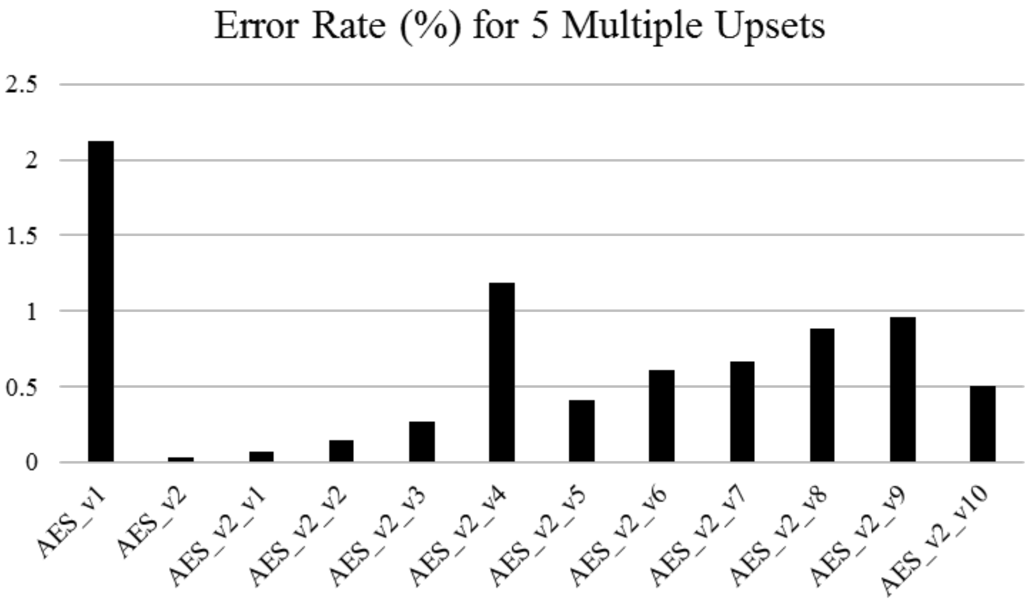

Fig. 43 Error rate for 5 multiple faults in TMR designs with different granularities

In terms of MVs, this result showed in figure 43 means that by removing the voters r1, r2 and even r3, it does not provoke a significant impact in terms of error rate for multiple faults, but it can save some area. On the other hand, the AES version AES_v2_v7 has show a high error rate for double and triple faults. This means that voter _r6 seems to be important to those types of faults. When 4 and 5 multiple faults were injected, the AES_v2_v4 presents a high error rate. This suggests that voter _r4 has an important effect on mitigating multiple faults. Note that when MV _r4 is removed (AESv2_v4), the error rate increases approximately in 441% to multiple 5 upsets in comparison the previous circuit (AESv2_v3). Voter MV r4 is the voter located at the input of the mux block before AddKey. It seems that for this specific design, the inputs of the Addkey block are crucial for multiple upsets, and the use of voters in those cases increases the probability of multiple upsets to overcome the TMR design. However, when removing voter r10 in the AES_v2_v5, the error rate drops again. Therefore, it seems that removing both voters *r4 and r*10 is better than only removing _r4.

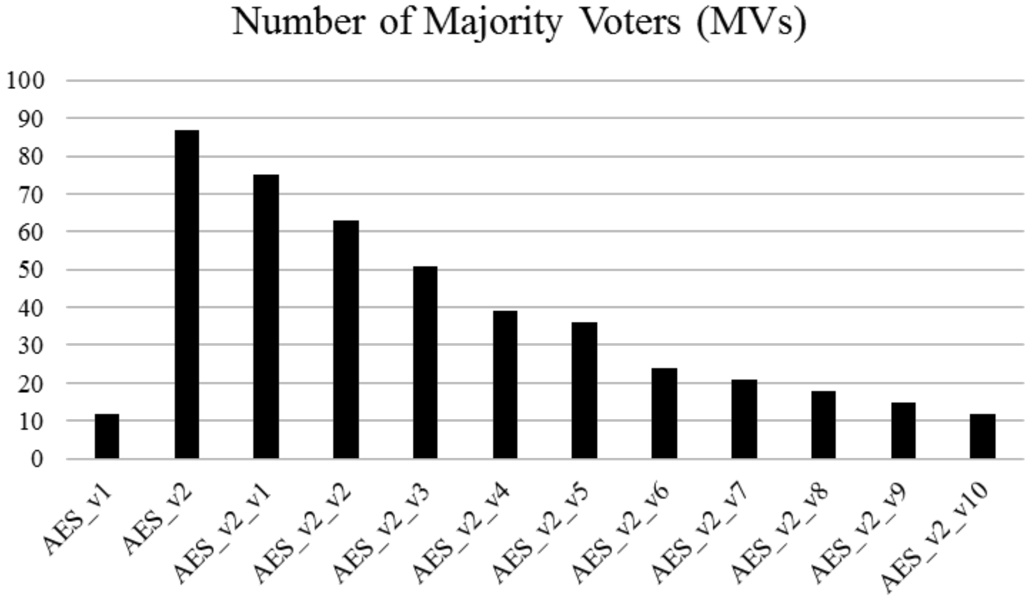

Fig. 44 Number of Majority Voters (MVs) in TMR designs with different granularities

Note that although MVs aims to increase the probability of mitigating faults, they also increase the probability of bringing together cells that are highly related to each other from distinct redundant blocks of the TMR. Consequently, there is a limitation on multiple fault tolerance even when adding a large number of MVs. Therefore, even the AES_v2 cannot cope with all the triple faults and multiple 4 and 5 faults. In this case, only specific placement and the increase of space between the cells can reduce the SER.

Figures 44 to 45 show the plotted results of the number of voters, area and performance respectively. It is interesting to observe that AES solution AES_v2_v3 may present one good trade-off option for error rate and area, and AES_v2_v5 with a slither higher error rate but with even more reduced area.

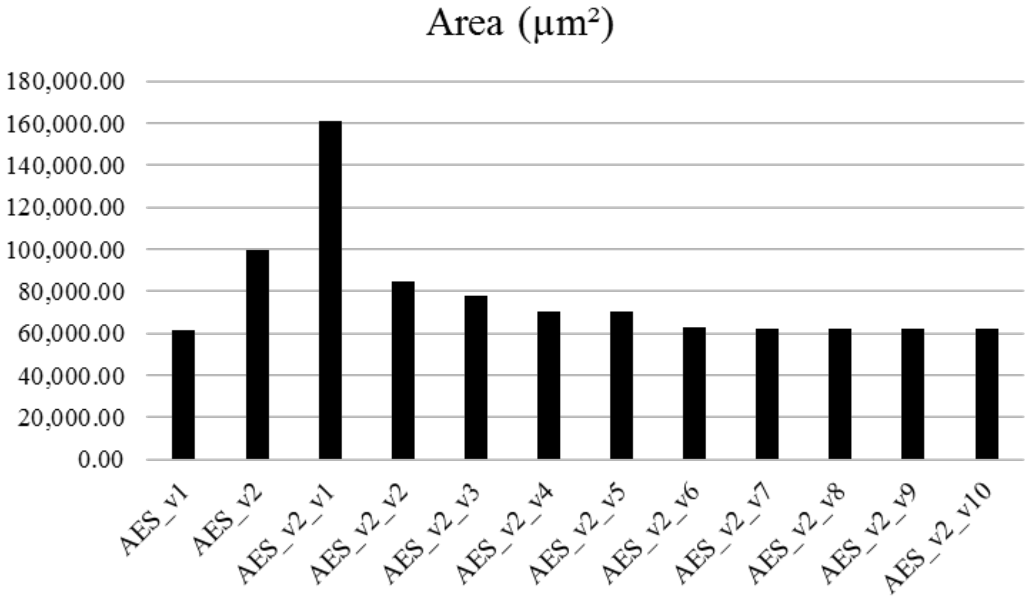

Fig. 45 Area in TMR designs with different granularities

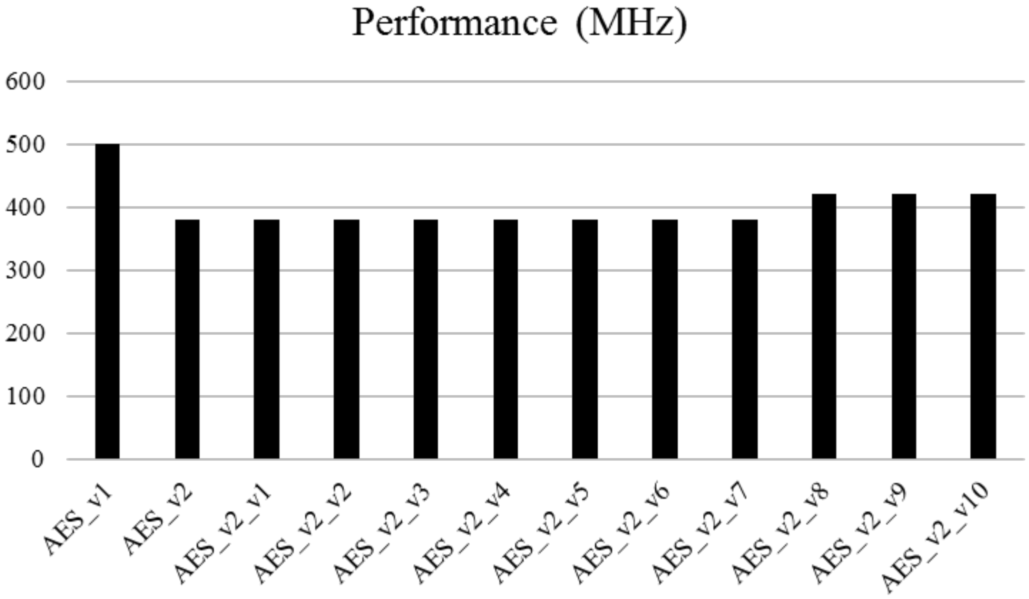

Fig. 46 Performance in TMR designs with different granularities

The effects of single-event-induced charge sharing were investigated in several TMR schemes with different levels of granularities (ALMEIDA, 2012). Results have shown that multiple upsets can easily overcome the robustness of the TMR. Increasing the number of majority voters (MV) leads to a reduced soft error rate under multiple faults. However, there are also interesting tradeoffs between number of MVs, and consequently area, and the SER for multiple faults.

7 EXPLORING DIFFERENT GRANULARITIES OF FINE GRAIN TMR (FGTMR) TO COPE WITH MULTIPLE FAULTS INS SRAM-BASED FPGAS

System designs operating in high reliability applications, such as particles accelerators, satellites and aircrafts require high tolerance to errors as possible. However, many Commercial Off-The-Shelf (COTS) products have been employed in these critical areas in recent years. Adopting COTS brings benefit to the project as they include low cost hardware and software and they are widely available in the commercial market. On the other hand, COTS are usually very sensitive to radiation effects and efficient mitigation techniques must be employed to reduce SER and increase the fault tolerance. In this context, reconfigurable architectures such as SRAM-based FPGAs have gained more and more attention over the past years.

State-of-the-art SRAM-based FPGAs present a set of features that are relevant for systems operating in high reliability applications, such as flexibility, high performance and fast time-to-market. However, the configuration memories of commercial SRAM-based FPGAs are usually based in standard SRAM cells (ITRS, 2014), which are very susceptible to SEUs.

SRAM-based FPGAs are composed of an array of Configurable Logic Blocks (CLB), a complex routing architecture, an array of embedded memories (Block RAM), an array of Digital Signal Processing components (DSP) and a set of control and management logic. The CLBs are composed of a Look-up Table (LUT) that implements the combinational logic and Flip-Flops (DFF) that implement the sequential elements. The routing architecture can be very complex and composed of millions of pre-defined wires that can be configured by multiplexers and switches to build the desired routing.

The configuration of all CLBs, routing, Block RAMs, DSP blocks and I/O blocks is done by a set of configuration memory bits called bitstream. According to the size of the FPGA device, the bitstream can contain millions of bits. In modern FPGAs the bitstream is divided into frames to allow partial reconfiguration. The memory bits that store the bitstream inside the FPGA are composed of SRAM memory cells, so they are reprogrammable and volatile. When an SEU occurs in a configuration memory bit of an SRAM-based FPGA, it can provoke a bit-flip. This bit-flip can change the configuration of a routing connection or the configuration of a LUT or flip-flop in the CLB. This can have severe repercussions in the designed circuit, since an SEU may change its functionality.

A SEU in the configuration memory bits of an SRAM-based FPGA has a persistent effect and it can only be corrected with the load a correct bitstream. In the combinational logic, an SEU cause a persistent fault in one or more configuration bits of a LUT, changing its truth table. SEU in the routing architecture can connect or disconnect a wire in the matrix modifying the mapped circuit. A high number of clock cycles may be required to have the persistent error detected and initiate recovery actions such as the load of a fault-free bitstream. During this latency, the error can propagate to the rest of the system. Bit-flips can also occur in the flip-flop of the CLB used to implement the user's sequential logic. In this case, the bit-flip has a transient effect and a load of the flip-flop will correct it.

In this way, the main contribution of this dissertation is evaluating the robustness of several TMR schemes with different levels of granularity in SRAM-based FPGAs aiming to establish a relation between TMR granularity levels and their used resources versus fault tolerance to multiple faults due to charge sharing or accumulation of faults. In addition, to analyze how much the insertion of MVs can really impart the cross-section and the number of accumulated upsets in the bitstream before the design fails.

The same case-study circuit AES was used. The circuit was evaluated under fault injection and under neutrons at LANSCE, Los Alamos, USA. The circuits were exposed to a mean neutron particles flux of 3.98x10^4 n/cm²/s with energies above 10 MeV during 1,268 minutes, which resulted into an amount of fourteen trials. Then, calculated the fluence that is the number of particles passed through cm^2. The observed SEU rate is calculated in terms of static cross-section, dynamic cross-section and Failure in Time (FIT).

When performing radiation tests, the results are also analyzed in according to the error rate, in this case due to the number of particles that pass the design during a certain time. For SRAM-based FPGA, the error rate is shown by calculating the cross-section and Failure in Time (FIT).

Static cross-section is the probability that a particle generate a SEU during the experiment, given in cm^2/device. For example, a cross-section of 5x10^{-7} cm^2 means that it is necessary 1/5x10^{-7} = 2x10^6 particles passing by the device to cause one SEU in the configuration (TARRILLO, 2014). The static cross-section is expressed for number of SEE divided by fluence (1), and it can be describing on function of number of bits (2).

$$\sigma_{static} = \frac{number\ of\ SEEs\ recorded\ during\ the\ experiment}{fluence\ of\ the\ experiment} \quad (1)$$

$$\sigma_{bit} = \frac{number\ of\ SEEs\, recorded\ during\ the\ experiment}{fluence\ \ of\ the\ experiment\ x\ number\ of\ bits} \qquad (2)$$

Dynamic cross-section is the probability that a determined particle generate an error in the circuit, given in cm^2/device. For example, a cross-section of 1.32x10^{-8} cm^2 means that it is necessary 1/1.32x10^{-8}=7.58x10^7 particles passing by the device to cause one error in the output. It is described in equation (3).

$$\sigma_{dynamic} = \frac{number\ of\ SEEs\, recorded\ during\ the\ experiment}{fluence\ of\ the\ experiment} \qquad (3)$$

Second, it is possible to find the error rate in terms of FIT, for example. The FIT defines the expected number of errors in 10^9 hours. Therefore, a circuit with lower cross-section and consequently a lower FIT is more robust to faults than a circuit that presents a higher cross-section and FIT under the same environment conditions.

The cryptographic core that implements the AES algorithm has 128-bit key and data. In order to evaluate the neutron-induced effects in TMR schemes with different levels of granularities, four different TMR-AES schemes were designed. The first one in figure 46, AES_v1, defined as the Coarse Grain TMR (CGTMR). The MVs vote each signal bit-to-bit.

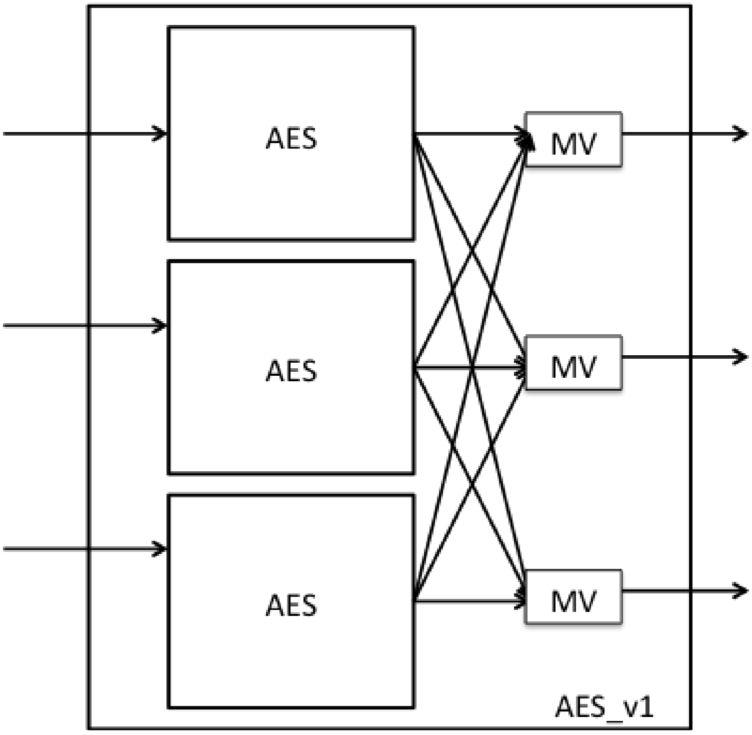

Fig. 47 AES circuits protected by TMR with large granularity (AES_v1)

The figure 47 shows the AES_v2 that divide the AES logic into small TMR logic blocks acting as a Fine Grain TMR (FGTMR).

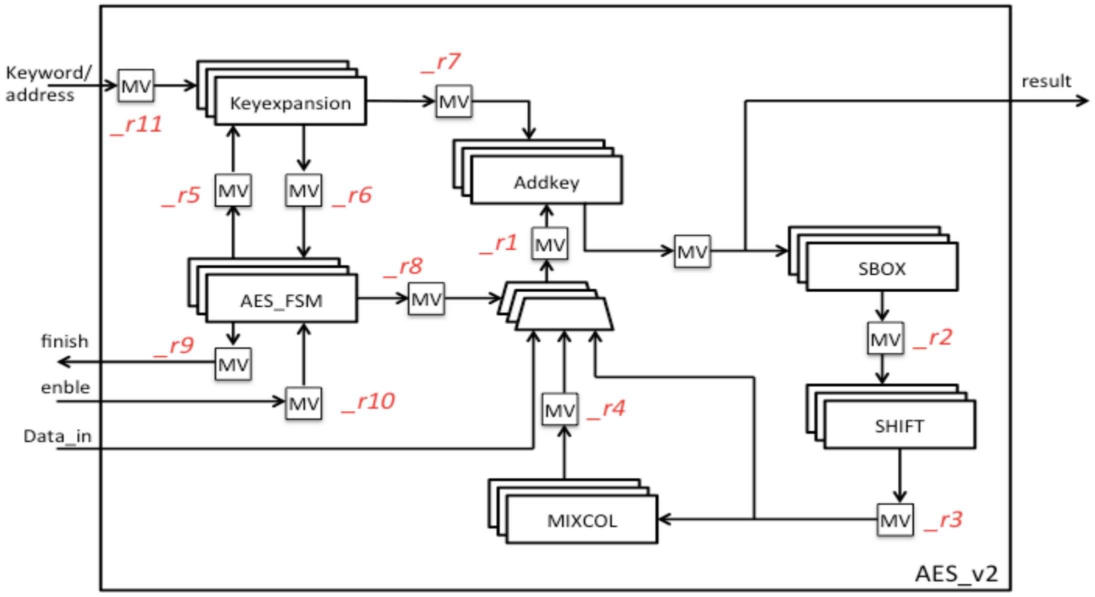

Fig. 48 AES circuits protected by TMR with small granularity (AES_v2)

From this second design, two slightly different designs were generated. AES_v3 and AES_v4 remove different MVs located at different positions in the design, as described in table XII.

Tab. IX Different AES circuit with the correspondent removed MVs

Circuit	Removed MVs
AES_v2	None
AES_v3	_r1
AES_v4	_r1, _r2, _r3, _r4, _r7, _r10

The AES circuits were prototyped in a Xilinx Spartan-6 LX45 SRAM-based FPGA (WANG, 2010) with an input frequency of 100 MHz Synthesis results are shown in Tab. XIII. AES_v1 has a total of 5,067 LUTs (3.23 times larger than the non-protected version), while AES_v2 has a total of 9,287 LUTs (5.92 times larger than the non-protected version). In terms of performance, the AES_v1 presents a reduction of 0.3% in speed compared to the non-protected design, while AES_v2 presents a reduction of 34% due to the large number of MVs inserted. As it is possible to observe, the addition of MVs has a dramatic impact in terms of area and performance of the design.

Tab. X FPGA resources occupation and performance of the TMR-AES schemes

AES Designs	LUTs	MUXs	FFs	DSPs	Resource overhead (%)	Critical bits	Number of MVs
AES_v1	5067	312	2857	0	444.95	421541	12
AES_v2	9287	312	2857	0	672.93	696800	87
AES_v3	8318	312	2857	0	620.58	611868	75
AES_v4	5639	312	2857	0	475.85	477906	24

Figure 48 and table XIV show the calculated dynamic cross sections from the AES. The graphs analysis enables us to conclude that by increasing the number of MVs, the SER is normally reduced. Results show that a FGTMR (_V2) can reduce from 65% (TMR-AES) the FIT of a system when compared to a CGTMR approach. Clearly, the second version of both case studies circuits, which present the highest number of MVs (and consequently the largest area and the worst performance), have the reduced dynamic cross section. If MVs are removed, in general there is a trade-off that must be analyzed in terms of SER, area and performance. Please note that the difference in terms of cross section varies 2.94 times in AES case.

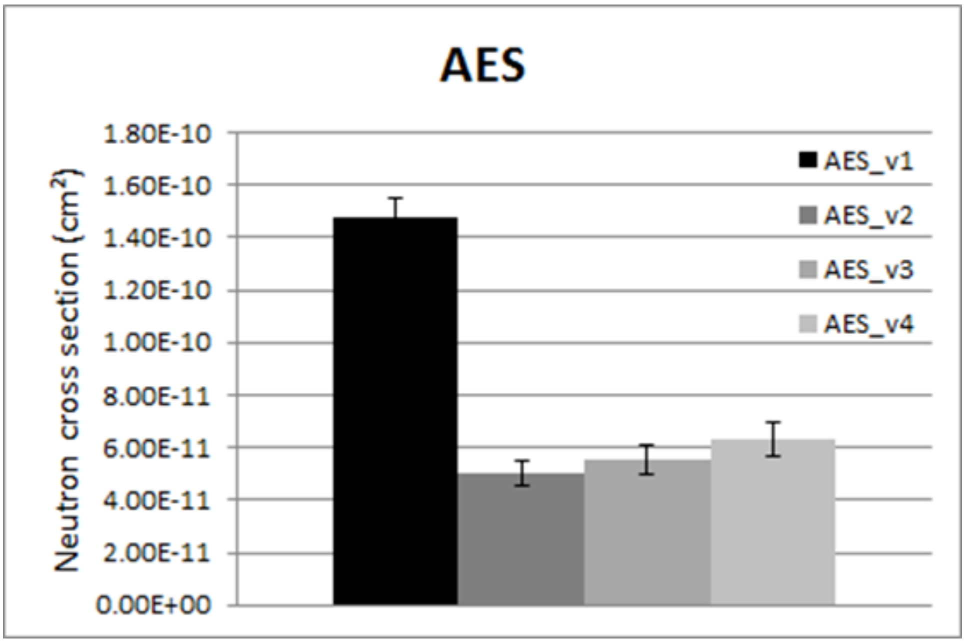

Fig. 49 Calculated dynamic cross sections for each version of the AES case study during the neutron experiment

Tab. XI Obtained dynamic cross-sections and FITs for the different AES designs

AES Designs	Dynamic cross section (cm²)	FIT
AES_V1	1.47×10^{-10}	1.91
AES_V2	5.01×10^{-11}	0.65
AES_V3	5.56×10^{-11}	0.72
AES_V4	6.34×10^{-11}	0.82

Regarding the SEU accumulation impacts in the configuration memory of the case-study circuits, a comparison of its effects among the different versions of each circuit is shown in figure 49. As data show, in the design with a fine grain scheme (FGTMR) were more effective in masking SEU accumulation in the configuration memory when compared to a coarse grain scheme.

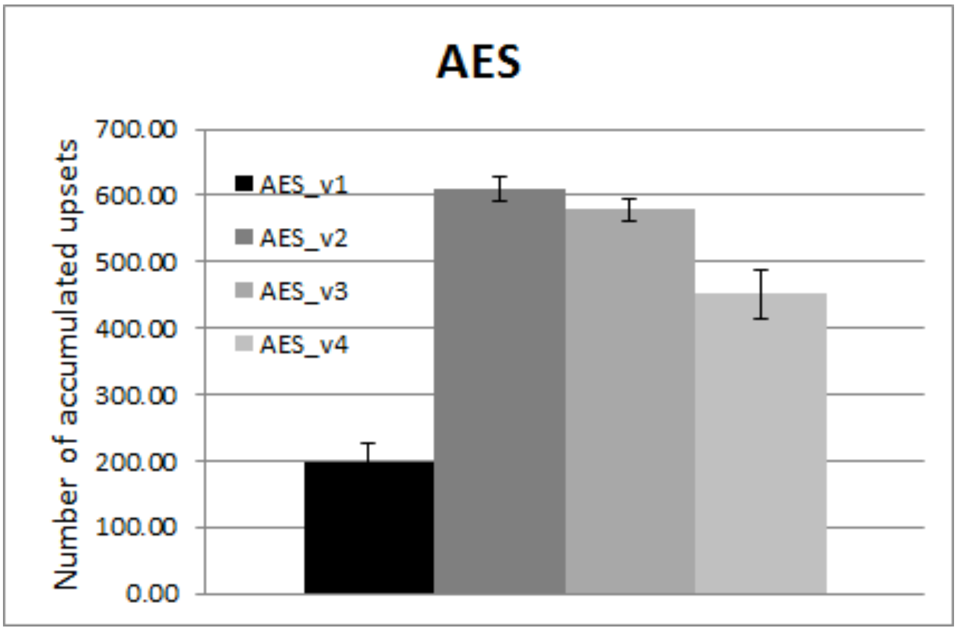

Fig. 50 SEU accumulated effects observed in the configuration memory bits of the AES case study during the neutron experiment

It is important to highlight that in the case-study circuit, the intermediate versions presented a slight difference among them in terms of fault tolerance, which means that if resource usage is a concern, a good trade-off could be achieved with a scheme with an intermediate grain scheme.

Evaluating the robustness of a set of TMR schemes with different levels of granularity in SRAM-based FPGAs aiming to establish a relation between TMR granularity levels and their used resources versus fault tolerance. Results have shown that increasing the number of majority voters leads to a reduced soft error rate and to achieve a higher fault tolerance level. However, results also show that there are important trades-off among number of majority voters, area and performance.

8 CONCLUSION

Charge sharing is a significant SEE issue that must be properly addressed. This work proposes a charge sharing evaluation methodology that uses the placement information to create a fault injection list that correlates better with the actual physical behavior. Also, the fault injection campaigns were accelerated by means of emulation. This allows the methodology to be used for evaluating more complex devices where simulation alone would not be feasible.

The results of the injection campaigns showed that large overestimations might occur unless placement data is considered. On the other hand, the error rate for single-event induced charge sharing strongly depends on the sensitivity of neighbor cells. In comparison with single SET effects, single-event-induced charge sharing can provoke large variations, ranging for a large increase to even a slight decrease of the error rate. The methodology proposed in this work provides a solution to identify the most critical nodes to be hardened in order to mitigate single-event-induced charge sharing effects in complex circuits.

The effects of single-event-induced charge sharing were investigated in several TMR schemes with different levels of granularities. Results have shown that multiple upsets can easily overcome the robustness of the TMR. Increasing the number of majority voters (MV) leads to a reduced soft error rate under multiple faults. However, there are also interesting tradeoffs between number of MVs, and consequently area, and the SER for multiple faults.

Mitigating the effects of single-event-induced charge sharing is necessary as these effects are becoming critical for advanced technologies. In this work, we have demonstrated that placement may have a significant impact in the soft error rate due to charge sharing and have proposed a logic cell placement optimization approach to minimize single-event-induced charge sharing effects in integrated circuits. This approach is based on a powerful analysis of the SER for every possible pair of nodes and the generation of a set of additional constraints to be used during the placement process. With this analysis, we have also shown that charge sharing effects are not always negative. Actually, for every node in the case study circuit we were able to identify other nodes that can reduce the SER due to double faults even below the SER due to single faults.

Integrated circuits placement tools use complex algorithms to find an optimal balance among multiple constraints, typically related to area and timing optimization. The results show that the SER due to charge sharing can be reduced by using additional mitigation-driven constraints, even though only a subset of them were honored in our experiments using a commercial placement tool. Future work emphasizes the development of mitigation-driven placement algorithms, which prioritize mitigation-driven constraints. Additional reductions may also be obtained by combining the proposed constraining methodology with other techniques, such as nodal spacing or node duplication.

REFERENCES

Advanced Encryption Standard (AES), National Institute of Standards and Technology (NIST) [Online].Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

AHLBIN, J.R. et al. Single-event transient pulse quenching in advanced CMOS logic circuits. **IEEE Transactions on Nuclear Science**, v. 56, n. 6, 2009. p. 3050-3056

ALMEIDA, F. et al. Single-event-induced charge sharing effects in TMR with different levels of granularity. In: RADIATION AND ITS EFFECTS ON COMPONENTS AND SYSTEMS (RADECS), 2012…**Proceedings,** Biarritz: IEEE, 2012

AMUSAN, O.A. et al. Analysis of single event vulnerabilities in a 130 nm CMOS technology. **M.S. thesis**, Nashville, 2006.

AMUSAN, O.A. et al. Charge collection and charge sharing in a 130 nm CMOS technology. **IEEE Transactions on Nuclear Science**, v. 53, n. 6, 2006. p. 3253-3258

ATKINSON, N.M. et al. Layout technique for single-event transient mitigation via pulse quenching. **IEEE Transactions on Nuclear Science**, v. 58, n.3, 2011. p. 885-890

BIWEI L. et al. Temperature dependency of charge sharing and MBU sensitivity in 130-nm CMOS technology. **IEEE Transactions on Nuclear Science**, v. 56, n. 4, 2009. p. 2473-2479

BLACK, J.D. et al. HBD layout isolation techniques for multiple node charge collection mitigation. **IEEE Transactions on Nuclear Science**, v. 52, n .6, 2005. p. 2536-2541

BLUM, D.R. et al. Hardened by design techniques for implementing multiple-bit upset tolerant static memories. In: INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), 2007…**Proceedings**, [S.l.]: IEEE, 2007. p. 2786-2789

BOHR, M. et al. The evolution of scaling from the homogeneous era to the heterogeneous era. In: INTERNATIONAL ELECTRON DEVICES MEETING (IEDM), 2011…**Proceedings** [S.l.]: IEEE, 2011. p. 5-7

BORGES, G. et al. Diversity TMR: Proof of concept in a mixed-signal case. In: LATIN AMERICAN TEST WORKSHOP (LATW), 2010…**Proceedings**, [S.l.:s.n.], 2010. p.1-6

DI FEDERICO, M. et al. Integrated circuit implementation of multi-dimensional piecewise-linear functions. **Digital Signal Process**, v. 20, 2010. p. 1723–1732

DIEHL-NAGLE, S.E. et al. Single event upset rate predictions for complex logic systems. **IEEE Transactions on Nuclear Science**, v. 31, n. 6, 1984. p. 1132-1138

DODD, P.E. et al. Basic mechanisms and modeling of single-event upset in digital microelectronics. **IEEE Transactions on Nuclear Science**, v. 50, n. 3, 2003. p. 583-602

ENTRENA, L. et al. Soft error sensitivity evaluation of microprocessors by multilevel emulation-based fault injection. **IEEE Transactions on Computers**, v. 61, n. 3, 2012. p. 313-322

ENTRENA, L. et al. Constrained placement methodology for reducing SER under single-event-induced charge sharing effects. **IEEE Transactions on Nuclear Science**, v. 59, n. 4, 2012. p. 811-817

ENTRENA, L. et al. SET emulation considering electrical masking effects. **IEEE Transactions on Nuclear Science**, v. 56, n. 4, 2009. p. 2021-2025

FERLET-CAVROIS, V. et al. Direct measurement of transient pulses induced by laser and heavy ion irradiation in deca-nanometer devices. **IEEE Transactions on Nuclear Science**, v. 52, n. 6, 2005. p. 2104- 2113

GIOT, D. et al. Heavy ion testing and 3-D simulations of multiple cell upset in 65 nm standard SRAMs. **IEEE Transactions on Nuclear Science**, v. 55, n. 4, 2008. p. 2048-2054

HAN, J. et al. On the reliability of computational structures using majority logic. **IEEE Transactions on Nanotechnology**, v. 10, n. 5, 2011. p. 1099-1112

HARADA, R. et al. Neutron induced single event multiple transients with voltage scaling and body biasing. **IEEE International Reliability Physics Symposium**, [S.l.:s.n.], 2011. p. 3C.4.1-3C.4.5, 10-14

IROM, F. et al. Investigation of single-event transients in linear voltage regulators. **IEEE Transactions on Nuclear Science,** v. 55, n. 6, 2008. p. 3352-3359

VON NEUMANN, J.V. et al. Probabilistic logics and synthesis of reliable organisms from unreliable components. **Automata Studies**. [S.l.]: Princeton University Press, 1956. p. 43-98

KASTENSMIDT, F.L. et al. On the optimal design of triple modular redundancy logic for SRAM-based FPGAs. In: DESIGN, AUTOMATION AND TEST IN EUROPE (DATE), 2005…**Proceedings**, [S.l.], v. 2, n. 7-11, 2005. p. 1290-1295

LABEL, K.A. et al. Commercial microelectronics technologies for applications in the satellite radiation environment. In: AEROSPACE APPLICATIONS CONFERENCE (AAC), 1996…**Proceedings**, [s.n.], v. 1, p. 375-390

MAESTRO, J. A. et al. Study of the effects of MBUs on the reliability of a 150 nm SRAM device. In: DESIGN AUTOMATION CONFERENCE (DAC), 2008…**Proceedings**, [S.l.:s.n.], p.930-935

MANUZZATO, A. et al. Effectiveness of TMR-based techniques to mitigate alpha-induced SEU accumulation in commercial SRAM-based FPGAs. In: RADIATION AND ITS EFFECTS ON COMPONENTS AND SYSTEMS (RADECS), 2007…**Proceedings**, [S.l.:s.n.], 2007. p. 1-7

MASSENGILL, L.W. et al. Soft-Error Charge-Sharing Mechanisms at Sub-100nm Technology Nodes. In: INTEGRATED CIRCUIT DESIGN AND TECHNOLOGY, [S.l.:s.n.], 2007. p. 1-4

MESSENGER, G.C. et al. Collection of Charge on Junction Nodes from Ion Tracks. **IEEE Transactions on Nuclear Science**, v. 29, n. 6, 1982. p. 2024-2031

NASA. **Thesaurus**: subject terms for indexing scientific and technical information. [S.l.]: NASA, 1967. 3v

NEUBERGER G. et al. A multiple bit upset tolerant SRAM memory. **ACM Transactions on Design Automation of Electronic Systems**, v. 8, n. 4, 2003. p. 577-590

NIKNAHAD, M. et al. Fine grain fault tolerance – A key to high reliability for FPGAs in space. In: AEROSPACE APPLICATIONS CONFERENCE (AAC), 2012…**Proceedings**, [S.l.:s.n.], 2012. p. 1-10

O'BRYAN, M.V. et al. Recent radiation damage and single event effect results for microelectronics. In: RADIATION EFFECTS DATA WORKSHOP (REDW), 1999…**Proceedings**, [S.l.:s.n.], 1999. p.1-14

OLSON, B.D. et al. Simultaneous single event charge sharing and parasitic bipolar conduction in a highly-scaled SRAM design. **IEEE Transactions on Nuclear Science**, v. 52, n. 6, 2005. p. 2132-2136

OpenCores.org, **Avalon AES IP**, 2009 [Online]. Available: http://opencores.org/project

PAGLIARINI, S. et al. Analyzing the Impact of Single-Event-Induced Charge Sharing in Complex Circuits. **IEEE Transactions on Nuclear Science**, v. 58, n. 6, 2011. p. 2768-2775

PRADHAN, D. **Fault-tolerant computer system design**. New Jersey: Prentice-Hall, 1996

RADAELLI, D. et al. Investigation of multi-bit upsets in a 150 nm technology SRAM device. **IEEE Transactions on Nuclear Science**, v. 52, n. 6, 2005. p. 2433-2437

ROSSI, D. et al. Multiple transient faults in logic: an issue for next generation ICs?. In: DEFECT AND FAULT TOLERANCE IN VLSI SYSTEMS (DFTVS), 2005…**Proceedings**, [S.l:s.n.], 2005. p. 352-360

SAMUDRALA, P.K. et al. Selective triple Modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs. **IEEE Transactions on Nuclear Science**, v. 51, n. 5, 2004. p. 2957-2969

SIMEVSKI, A. et al. Scalable design of a programmable NMR voter with inputs' state descriptor and self-checking capability. Adaptive Hardware and Systems NASA/ESA Conference, 2012…**Proceedings**, [Sl.:s.n.], 2012. p. 182-189

Synopsys Armenia Educational Department, **SAED 90 nm generic library**, 2012 [Online]. Available: http://www.synopsys.com/Community/UniversityProgram

Synopsys Design Compiler. [Online]. Available: http://www.synopsys.com/

Synopsys, **IC compiler datasheet**, 2009 [Online]. Available: http://www.synopsys.com/Tools/Implementation/PhysicalImplementation/Documents/ iccompiler_ds.pdf

TAMBARA, L.A. et al. Decreasing FIT with diverse triple modular redundancy in SRAM-based FPGAs. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, 2014…**Proceedings**, [S.l.]: IEEE, 2014. p. 153-158

TAMBARA, L.A. et al. Evaluating the effectiveness of a diversity TMR scheme under neutrons. Radiation and Its Effects on Components and Systems (RADECS), 2013…**Proceedings**, [S.l.:s.n.], 2013. p.1-5

TARRILLO, J. Exploring the Use of Multiple Modular Redundancies for Masking Accumulated Faults in SRAM-based FPGAs", **Thesis (Ph.D.)**, PPGC of the UFRGS, 2014

VELAZCO, R. et al. SEU-hardened storage cell validation using a pulsed laser. **IEEE Transactions on Nuclear Science**, v. 43, n. 6, 1996. p. 2843-2848

VELAZCO, R. **Radiation effects on embedded systems**. Dordrecht: Springer, 2007. p. 1-9

WANG, X. et al. Partitioning Triple Modular Redundancy for Single Event Upset Mitigation in FPGA. In INTERNATIONAL CONFERENCE E-PRODUCT E-SERVICE AND E-ENTERTAINMENT, 2010…**Proceedings** [S.l.:s.n.], 2010. p. 1-4

YANKANG, D. et al. Impact of pulse quenching effect on the soft error vulnerabilities in combinational circuits based on standard cells. **Elsevier Microelectronics Journal**, v. 44, 2013. p. 65-71