UNIVERSITÀ DI PADOVA FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria dell’Informazione

Scuola di Dottorato di Ricerca in Ingegneria dell’Informazione

Indirizzo: Scienza e Tecnologia dell’Informazione

CICLO XXII

SOFT ERRORS INDUCED BY NEUTRONS AND ALPHA PARTICLES IN SYSTEMS ON CHIPS

Direttore della Scuola: Ch.mo Prof. Matteo Bertocco

Supervisore: Ch.mo Prof. Alessandro Paccagnella

Dottorando: Ing. Paolo Rech

“Don’t walk in front of me, I may not follow
Don’t walk behind me, I may not lead
Walk beside me and be my friend"

Albert Camus

If you don’t shoot for the moon,
you’ll never know what can be yours

Sommario

Questa tesi presenta un innovativo setup a basso costo per effettuare dei test sotto radiazione di System on Chips in cui siano integrati moduli di diversa natura e con diverse funzionalità. In particolare sono stati svolti numerosi test sotto radiazione di memorie SRAM integrate, di moduli logici integrati e di microprocessori integrati, analizzando i diversi protocolli di test necessari per poter caratterizzare al meglio la loro sensibilità alla radiazione.

Uno dei problemi maggiori che si riscontrano quando si deve testare un System on Chip è la ridotta accessibilità dei vari moduli integrati e i vincoli fisici che devono essere rispettati per effettuare il test stesso e che rendono le procedure di analisi molto difficili. I costruttori, per riuscire a verificare la funzionalità dei vari moduli integrati, usano molto spesso delle tecniche chiamate Design for Testability bastate su strutture di test integrate che permettono un'esaustiva verifica della funzionalità dei moduli minimizzando allo stesso tempo i costi del test. Durante gli esperimenti presentati in questo lavoro abbiamo riutilizzato alcune strutture integrate del tipo Design for Testability per caratterizzare nel dettaglio sia tutti i singoli moduli che compongono un System on Chip che il comportamento globale del dispositivo quando viene esposto a

radiazione. La strategia che è proposta in questa tesi può essere generalizzata e applicata a qualunque tipo di modulo integrato e sono presentati anche alcuni suggerimenti sul come applicare le strutture di test DfT agli esperimenti di radiazione. Quando si effettua un esperimenti di radiazione tipicamente ci sono diversi vincoli che, in base al laboratorio in cui gli esperimenti vengono eseguiti, possono essere imposti al setup di test. La scheda di test che abbiamo sviluppato ha una forma monolitica, che la rende facile da posizionare nella maggior parte delle camere di irraggiamento degli acceleratori di particelle utilizzati per questo tipo di esperienze. Inoltre, grazie da un lato all'integrazione delle strutture di test nel System on Chip da caratterizzare e, dall'altro, ad una strategia d'interfaccia che si basa sia sul JTAG che sui Wrappers, i test possono essere eseguiti ad alta frequenza usando però solamente connessioni lente fra un PC e il dispositivo da testare, diminuendo così drasticamente il costo globale degli esperimenti.

Questa tesi mostra e discute i risultati ottenuti da molte campagne di esperimenti di radiazione su un System on Chip costruito in tecnologia CMOS a 90 nm da STMicroelectronics. Tale dispositivo è stato pensato e realizzato per essere parte di un complesso progetto automotive; ci siamo dunque focalizzati sulle problematiche derivanti dall'impatto che la radiazione terrestre può avere in questo dispositivo. Abbiamo quindi esposto il chip sia a flussi di neutroni che di particelle alfa. Grazie ai dati ottenuti dagli esperimenti, abbiamo calcolato la sensibilità del modulo SRAM sia a particelle alfa che a neutroni, e abbiamo scoperto che quest'ultima è decisamente inferiore della prima. Abbiamo quindi caratterizzato il comportamento del microprocessore quando è esposto a particelle alfa. Il test statico ha dimostrato che i flip-flop che costituiscono i registri interni del microprocessore hanno un tasso di errore indotto da radiazione più elevato rispetto al modulo memoria utente e memoria codice. Questo risultato è di grande importanza e deve essere considerato, per esempio, quando si costruisce una piattaforma di fault-injection. Per effettuare il test dinamico del microprocessore abbiamo costruito due diversi codici di riferimento, in modo da capire come la corruzione delle riverse risorse di memorizzazione influenzi l'esecuzione del codice. I risultati ottenuti dimostrano che, in una tipica applicazione, gli errori nella memoria codice sono decisamente predominanti rispetto a quelli nei registri interni. Inoltre abbiamo visto che i bit di memoria codice e dei registri non sono sempre critici, e la loro corruzione non necessariamente si propaga all'uscita. Infine, abbiamo considerato l'efficacia e i costi di diverse tecniche di irrobustimento. In particolare, abbiamo studiato come l'ottimizzazione del layout proposta del Design For

Manufacturing o la Triple Module Redundancy influenzino la sensibilità alla radiazione del microprocessore. Abbiamo considerato dei chip costruiti con diversi livelli di maturità del Design For Manufacturing e i risultati sperimentali dimostrano che un più alto livello di ottimizzazione aumenta la resistenza del dispositivo alla radiazione alfa. Le tecniche di irrobustimento, comunque, hanno un costo. La decisione su quale tecnica adottare quando si costruisce un dispositivo complesso è un trade-off fra costi, performance e, ovviamente, affidabilità. Le strategie da adottare per un particolare prodotto dipendono quindi dai suoi requisiti e dall'ambiente in cui dovrà essere impiegato.

Abstract

This Manuscript presents a new low-cost test setup for the radiation tests of System on Chips composed of different functional modules of different nature. Particular attention is given to radiation experiments results of embedded SRAM cores, embedded logic cores and embedded microprocessor cores, highlighting the dissimilar test protocols required to characterize their sensitivity to radiation.

The main issues when testing a System on Chip are the cores reduced accessibility and the physical constraints test facilities may impose to the test setup. Manufacturers heavily employ Design for Testability techniques, based on built-in test structures, to enable exhaustive devices testing while minimizing application costs. We reused some of the Design for Testability built-in structures to deeply characterize the cores composing the System on Chip and the overall chip behaviours when exposed to radiation. Our strategy can be applied to any kind of integrated core, and we also present some guidelines on how built-in structures may be fruitfully applied to radiation experiments. Moreover, the monolithic shape of our test board makes it easy to be mounted in most of available particle accelerators chambers or radiation test facilities. As the test structures are built-in and thanks to the efficient interfaces strategy that takes *advantage of both JTAG and Wrappers standards, tests are performed at high frequency, thus avoiding Single Event Transients underestimation, but without the need of high-speed connections between a host PC and the DUT, drastically reducing the overall setup costs.*

This thesis also shows and discusses the results gained during massive radiation experiments campaigns on the available System on Chip manufactured by STMicroelectronics in a 90 nm CMOS technology. As device is meant to be part of a complex automotive design, it may be affected by ground level radiation. We then exposed the chips both to neutron and alpha particles fluxes. With our low-cost setup we measured the SRAM core cross section to alphas and neutrons, and found out that the former one is higher than the latter. We have also characterized the microprocessors behaviour when exposed to alphas. The static test stated that registers flip-flops have a higher radiation induced error rate with respect to code and user RAM one. This result is of great importance, and should be taken into account when building a fault-injection platform. To understand how the corruption of the different memory resources affects codes executions, we designed different benchmark codes and performed a dynamic test. Results demonstrate that, in a typical application, the bit-flips in the code RAM are definitely predominant with respect to the ones in registers. Moreover, we show how code RAM and register bits are not always critical, and their corruption does not necessarily propagate to outputs. Finally, we have considered hardening techniques efficiency and costs. In particular, we have studied how Design For Manufacturing layout modifications and Triple Module Redundancy affect the radiation sensitivity of microprocessors. We considered chips built with different Design For Manufacturing maturity levels, and experimental results demonstrate that an higher level of optimization enhances the resilience to alpha radiation. Hardening techniques, however, come to a cost. The decision on which hardening technique to adopt when building a complex device is a hard-earned trade-off between costs, performance and, of course, reliability. Mitigation strategies for a product then depends on its requirements and on its mission environment.

Index

Chapter 1

Introduction

Today, radiation effects are a concern for electronics reliability and dependability not only in the space environment, but also at sea-level. For instance, the occurrence of Soft Errors (SEs) is due to neutrons originating from the interactions of cosmic rays with the atmosphere and even to alpha-emitting contaminants in the package/solder materials [Zie96][Dod02][Gas06]. Hence, radiation testing is becoming an important step in the qualification process, especially in the fields traditionally demanding high product reliability, such as the automotive or the biomedical one.

Modern highly-integrated System-on-Chips (SoCs) may be composed of up to hundreds of functional modules (cores) of different nature. The reduced accessibility of cores and the physical constraints to be respected make the test procedures very difficult. To enable exhaustive testing while minimizing application costs, Design for Testability (DfT) techniques, based on built-in test structures, are heavily employed.

This manuscript describes in details the testing structures typically used for Manufacturing tests of memory, logic, and microprocessor integrated cores. Then an efficient low-cost strategy for collecting data during radiation experiments on Systems-on-Chips is proposed, exploiting the available on-chip Design for Testability structures.

Radiation experiments were performed on a test vehicle built by STMicroelectronics in a 90 nm CMOS technology exposed to both neutrons and alpha particles. This introductory chapter gives a brief overview on radiation effects in digital electronics, on experiments that are typically performed to measure the radiation sensitivity of the devices, and on manufacturing test solutions. The second part of the chapter describes the motivations at the basis of this work and gives an outline of the manuscript.

1.1 Radiation Effects

Space level electronic devices are hit by ionizing particles coming from the cosmic rays and the solar wind [Mil25][Mey74]. In particular, the complex space radiation environments consist of particles trapped by planetary magnetospheres, including protons, electrons, and heavier ions, but also interplanetary particles which include protons and heavy ions of all the elements of the periodic table, and primary or secondary particles in the atmosphere of planets. At ground level radiation is still an issue, in fact both neutrons coming from the interaction of cosmic rays and atmosphere or particles generated by radioactive materials may disturb electronic devices.

The next paragraphs will give an overview on the different radiation environments, will describe how the different particles affect electronic components functionalities, and how devices are typically tested and characterized.

Figure 1.1: *Space level electronic devices are constantly hit by a great number of high energy ionizing particles*

1.1.1 Space and ground level radiation enviroments

The sun’s outer atmosphere, named the corona, continuously emits a stream of protons, electrons and a small amount of other ions, collectively called the solar wind. In addition to the solar wind, interplanetary space contains also high energy charged particles called cosmic rays, reaching energies of TeVs [Sma85][Cro97]. The solar activity may alter the space scenario, in fact during active phases of the solar cycle, the number and intensity of coronal mass ejections increases. These events can cause periodic increases in the level of interplanetary particles that are orders of magnitude higher than the cosmic rays environment.

The Earth is dipped in this scenario, and the interaction of its magnetic field and the solar wind formed a cavity named magnetosphere (Fig. 1.2). The presence of the solar wind gives to the magnetosphere a nearly symmetric about the magnetic axes shape, extending outward to long distances, and open at the poles. On the dayside, during moderate solar wind conditions, the solar wind plasma cannot penetrate deeply into the geomagnetic field because of its charged particle composition, and so the 99% of the solar wind particles passes around the Earth’s magnetosphere. The magnetosphere is filled with plasma that origins from the ionosphere and the solar wind. The plasmasphere is at low and mid latitudes in the inner magnetosphere, and the plasma

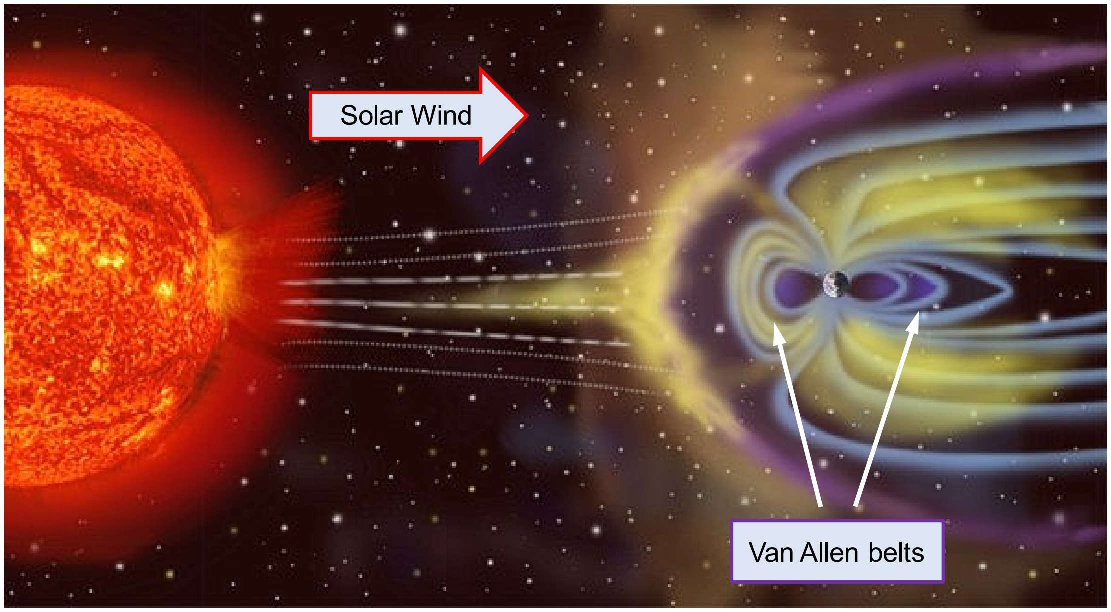

Figure 1.2: *the Earth’s magnetosphere generated by the solar wind and the Van Allen belts.*

sheet resides in the magnetotail. Overlapping the plasmashere and the plasma sheet are the high energy Van Allen radiation belts (fig. 1.2). The trapped electrons have energies up to tens of MeV, and the trapped protons and heavier ions up to hundreds of MeV [Gus96][Dag01].

The earth is then hit by elementary particles and atomic nuclei of very large energies coming from cosmic rays or solar wind. As stated above, most of them are protons (hydrogen nuclei) and all sorts of heavy ions. As galactic cosmic rays and solar wind particles enter the top of the Earth's atmosphere, they are attenuated by interaction with nitrogen and oxygen atoms. The result is that when a cosmic ray interact with our atmosphere, it generates a cascade of particles that may have enough energy to reach ground (Fig. 1.3 a). The primary cosmic ray will hardly ever hit the ground but will collide with nuclei of the air, usually several ten kilometers high, generating many new particles. Products of the cosmic ray shower are protons, electrons, neutrons, heavy ions, muons, and pions. Neutrons may have energies up to hundreds of MeV, and, as we will see, may generate sever problems to avionics, as the maximum of the number of particles composing the shower is reach at aircrafts altitude (Fig. 1.3 b).

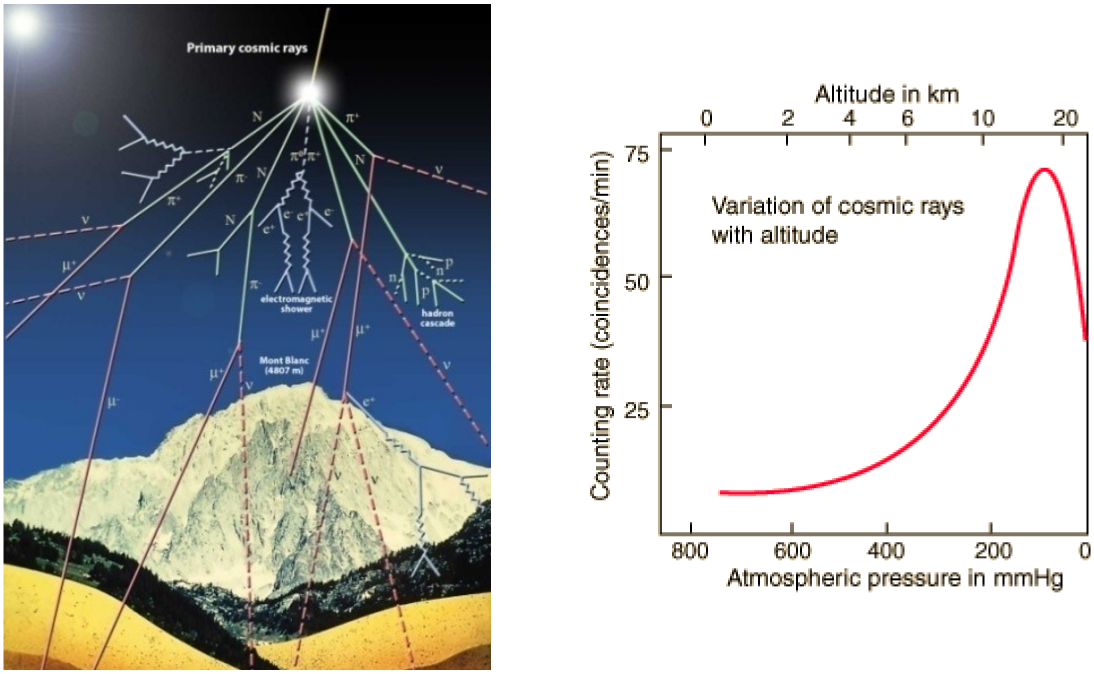

Figure 1.3: *a) on the left, the particles shower generated by a primary cosmic ray interaction with the terrestrial atmosphere. b) on the right, the shower reaches a maximum at avionics altitudes.*

Table 1.1: *Typical alpha particles fluxes for packaging materials and package type [Zie04]*

Material	**α particles flux (α/cm^2/h)**
Processed Wafers	0.0009
Cu Metal (thick)	0.0019
Al Metal (thick)	0.0014
Mold Compound	0.024 - 0.002
Underfill	0.002 - 0.0009
Pb-solders	7.2 - 0.002
Ceramic package	0.0011
Low Alpha (LA)	0.0002
Ultra Low Alpha (ULA)	0.0001

At ground level there may be also particles emitted by radioactive materials. Alpha particles, for instance, are high energy charged particles emitted from radioactive impurities in materials used in the chip package, such as solder balls or mold compound. These alpha particles have kinetic energies of the order of several MeV. Tab. 1.1 reports the typical alpha particles fluxes generated by different packaging materials and package types.

Electronic devices are then continuously exposed to radiation both at space level and at sea level. The interaction between impinging particles and the active area of electronic devices may generate different kind of errors and may have severe repercussions in the system functionalities. The mechanisms and effects of this interaction have been heavily studied since 1975, when Binder et al. reported the first soft fail from the analysis of satellite electronic [Bin75] [Zie04].

1.1.2 Soft errors

When a particle strikes an electronic device, it deposits (directly or indirectly) an amount of charge. If a high energy particle enters the substrate near the drain of a transistor (Fig. 1.4), it interacts with the substrate and causes many electron - hole pairs to be formed. The holes are quickly swept away to the bulk node, however the electrons are collected by the drain node. If the drain node is at a high voltage, these electrons will cause the voltage at the drain node to drop. The magnitude of the voltage drop depends

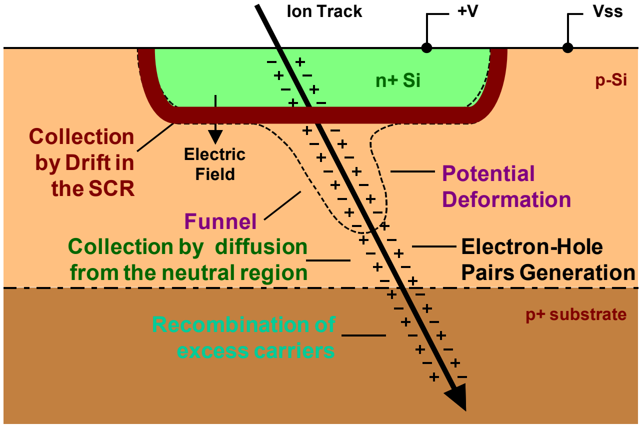
Ion Track
+V
Vss
n+ Si
p-Si
Collection by Drift in the SCR
Electric Field
Potential Deformation
Funnel
Collection by diffusion from the neutral region
Electron-Hole Pairs Generation
p+ substrate
Recombination of excess carriers

Figure 1.4: *Effects generated by the interaction of a particle with an electronic device.*

on the charge collected. If the amount of charge collected exceeds an amount known as the critical charge (Q_{crit}) an error will occur.

The most sensitive regions of the transistor are usually reversed biased p/n junctions, as the high field present in the reversed biased junction depletion region can be very efficient in collecting the particle induced charge through drift processes, leading to a transient current at the junction contact. Strikes near a depletion region can also result in significant transient currents as carriers diffuse into the vicinity of the depletion region field, where they can be efficiently collected. Moreover, the generated charge can locally collapse the junction electric field due to the highly conductive nature of the charge track and separation of charge by the depletion region. This funneling effect can increase charge collection at the struck node by extending the field deep in to the substrate. [Hsi81][Mcl82][Edm91].

There are two primary methods by which ionizing radiation releases charge in a semiconductor device: direct ionization by the incident particle itself and ionization by secondary particles created by nuclear reactions between the incident particle and the struck device. Both mechanism can lead to integrated circuits malfunction. On one side, when an energetic charged particle passes through a semiconductor material it frees electron hole pairs along its path as it loses energy. The energy loss per unit path length of a particle as it passes through a material is called Linear Energy Transfer (LET). On the other side, when a high energy proton or neutron enters the semiconductor lattice it may undergo an inelastic collision with a target nucleus. Possible nuclear reactions caused by this collision are Si recoils, emission of alpha or gamma particles and the recoil of a daughter nucleus, and spallation reactions, in which the target nucleus is broken into two fragments [Pet81][Wro00]. Any of these reaction products can deposit energy along their paths by direct ionization, as they are charged.

If some of the charge generated, directly or indirectly, by the interaction between the transistor and a single impinging particle is collected by a sensitive node of the device or circuit, and this charge is larger than the critical charge required to start an anomalous behavior, an effect named Single Event Effect may be seen, affecting the electrical performance of the device. The effects of these interaction between the impinging particle and the transistor may generate various kind of errors, depending on the hit device and particle energy or ionizing power. Radiation induced errors may be destructive errors (Hard Errors) or non-destructive errors (Soft Errors). Hard Errors are, for instance, Single Event Burnout, Single Event Gate Rupture, Stuck Bits, and Latchup. In any of the mentioned cases, the device functionality is permanently compromised, and typically device replacement (or, eventually annealing process) is the only applicable solution. Soft Errors, on the contrary, are generated by the radiation induced corruption of stored data or signals values, but without damaging the device.

In the case of memory elements, the charge deposited by the electronic device hitting particle may be enough to reverse the data state of a memory resource, generating a bit-flip. This kind of errors is traditionally addressed as Single Event Upset (SEU). With technology shrinking it may also be possible for one single particle to interact with more transistors, as the devices dimensions are becoming definitely smaller that the hitting particle track. Thus, one particle may corrupt more than one single bit, generating

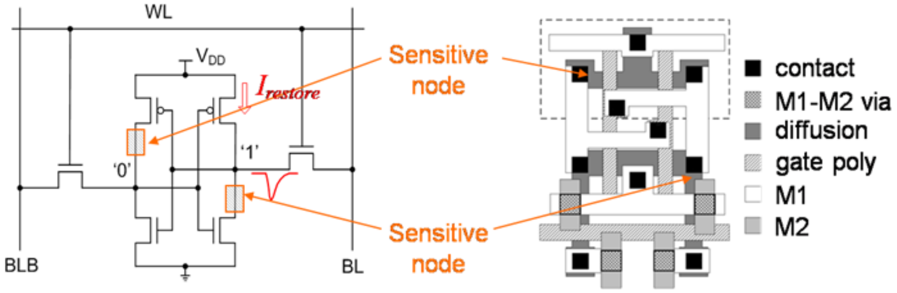

Figure 1.5: *Bit-flip generated in an SRAM cell by radiation induced charge deposition [CertiChip]*

a so-called Multiple Bit Upset (MBU).

When an energetic particle strikes a sensitive location in a SRAM (typically the reverse-biased drain junction of a transistor biased in the “off” state [Dod96][Wea88], as the “off” n-channel transistor shown in Fig. 1.5), charge collected by the junction results in a transient current in the struck transistor [Det97]. As this current flows through the transistor, the restoring transistor sources current to balance the particle-induced current that induces a voltage drop at its drain. This voltage transient is similar to a write pulse and can cause wrong memory state to be locked into the cell. The sensitivity of an SRAM cell to radiation depends on many factors, as the particle ionizing energy, the strike location, etc. The recovery time of the SRAM to a particle strike depends on the restoring transistor current drive [Axn86][Wea87]. The cell feedback time is simply the time required for the disturbed node voltage to feedback through the cross-coupled inverters and latch the wrong value. This time is related to the cell write time and can be thought of as the RC delay in the inverter pair. The smaller the RC delay, the faster the cell can respond to voltage transients and the more susceptible the SRAM is [Dod03].

Moving to a higher level of abstraction, the effects of the radiation induced data corruption may have dramatic effects, but also no effect, in the system functionalities. In fact, the data corrupted may be obsolete or unused, and once rewritten no remembrance of the Soft Error will remain. On the contrary, if the radiation corrupted data is going to be read and used, sever problems may occur and eventually compromise the system functionality. There are various Error Correction Code or other protection strategy to detect and eventually correct the radiation induced bit-flip. Unfortunately most of these strategies are designed to detect or correct at most one wrong bit per word, and the occurrence of Multiple Bit Upset corrupting more bit in the same word, may make them useless. Radiation-induced errors may be particularly severe in SRAM-based FPGAs, where modifications in the configuration memory [Bel04][Vio07] may alter the implemented circuits.

Moreover, combinatorial circuits are not immune to radiation as Single Event Transients (SET) may be generated [Bau02][Buc97][Zhu05]. A SET is a temporary voltage pulse at a struck node that may propagate and be latched in a memory element, leading to Soft Error [Bau05]. In a combinational circuit where the output is based on a logical relation to the inputs (with no capability for retention), if enough radiation-induced charge is collected, a short-lived charge is collected, and a short-lived transient in the output will be generated. If this radiation-induced “glitch” is actually propagated to the input of a latch or flip-flop during a latch clock signal, the erroneous input will be

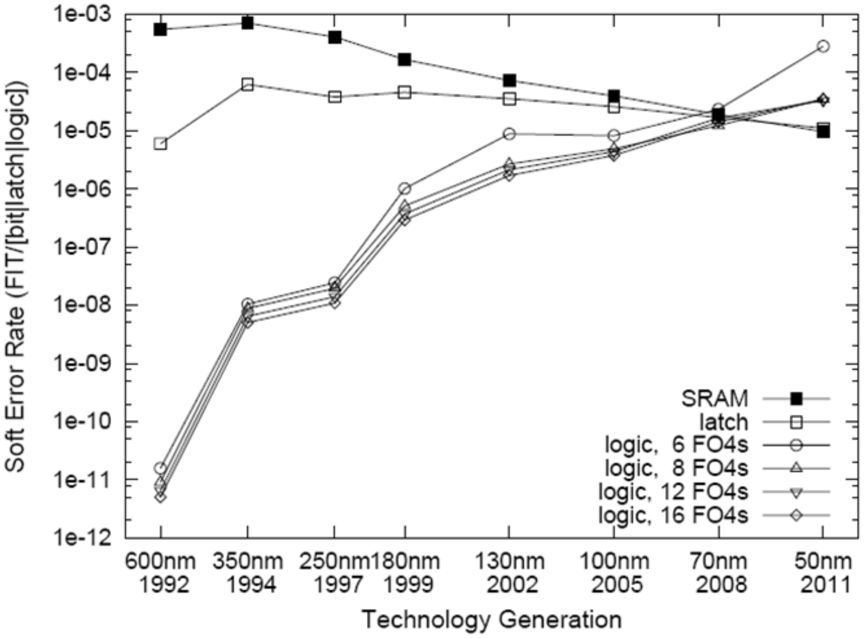

Figure 1.6: *Soft Error Rate of individual circuits [Shi02]*

"latched" and will be stored. We only consider a value to be stored in a latch if it is present and stable when the latch closes, since this value is passed to the next pipeline stage. A Soft Error occurs when the error pulse is stored into the level-sensitive latch at the end of a logic chain. It is also possible that a particle-induced pulse could delay the correct input signal from arriving at the latch input in time to be latched, thus causing an error. This type of error is referred to as delay fault which are, however, negligible in the current technologies. For older technologies, the SET could not propagate since it usually could not produce a full output swing and/or was quickly attenuated due to large load capacitances and large propagation delays. As reported in Fig. 1.6 [Shi02], the impact of SET in the overall system error rate is negligible with respect to SEU for technologies older than 70 nm. In advanced technologies where the propagation delay is reduced and the clock frequency is high, the SET can more easily traverse many logic gates, and its probability to be it is latched increases [Bau05].

As a transient change in the value of a logic circuit will not affect the results of computation unless it is captured in a memory circuit, as reported in [Shi02], transient error in a logic circuit might not be latched as it could be masked. Logical masking occurs when a particle strikes a portion of the combinational logic that does not affect the output due to a subsequent gate whose result is completely determined by its other input values. Electrical masking occurs when the pulse resulting from a particle strike is attenuated by subsequent logic gates due to the electrical properties of the gate to the point it does not affect the result of the circuit. Finally, latching-window masking occurs when the pulse resulting from a particle reached a latch, but not at the clock transition where the latch captures its input value. These masking effects could diminish significantly as feature sizes decrease and the number of stages in the processor pipeline increases. Electrical masking could be reduced by device scaling because smaller transistors are faster and therefore may have less attenuation effect on pulse. Also, deeper processor pipeline allow higher clock rates, meaning the latches in the processor will cycle more frequently, which may reduce latching-window masking. A pulse that is present at the latch input throughout the entire latching window will be latched and causes a SE (Fig. 1.7). When the pulse duration is lower that the size of the latching window, the probability of a SE is zero. On the contrary, when pulse duration exceeds the duration of an entire clock cycle and the size of the latching windows, it is assured to overlap at least one full latching window and hence has probability 1 of causing a SE. Working frequency has then a major impact on SETs capture: the higher the frequency the larger the probability of having a memory element corrupted by a propagating transient.

Many experiments have been performed and the data gained demonstrated how space and terrestrial applications reliability is seriously injured by radiation. Spacecrafts in orbit, for instance, visit regions outside of the Earth's magnetosphere where they are exposed to particles coming from the solar wind, and may works in literature report Single Event Upsets in satellites [Har90][Ada91]. Moreover, in the last solar cycle in October-November 2003, solar proton and heavy ions induced Single Event Transients

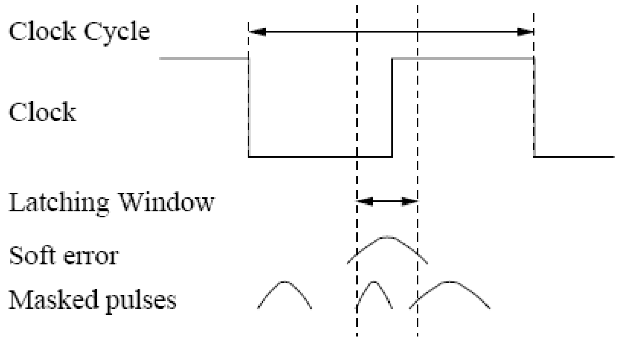

Figure 1.7: *Latching window masking [Shi02]*

were observed [Dye04]. The radiation hazard at avionics altitude is also of great danger. Below altitudes of about 60,000 feet, secondary neutrons from cosmic ray fragmentation are the most important contribution to SEUs [Tsa84], and several flight experiments [Nor96] have demonstrated that energetic particles can cause single event effects in electronics at avionic altitudes. Also at sea level radiation is an issue, in particular for the safety critical applications as the automotive and the biomedical ones. In fact, both high-energy and thermal neutrons generated by cosmic rays collision with terrestrial atmosphere and alpha particles emitted by chip and package materials may generate different kinds of errors in electronic devices. As J.F. Ziegler stated: "Soft Errors from radiation are the primary limit on digital electronic reliability. This phenomenon is more important that all other causes of computing reliability put together" [Zie04].

1.1.3 Radiation tests

Radiation tests aim at calculating the device sensitivity to different kinds of impinging particles. As device sensitivity is dependent on many operating factors such as voltage supply, frequency, temperature, etc., tests have to be performed in the full range of variation of these parameters. Sensitivity is calculated counting the number of Soft Errors as a function of fluence (device hitting particles per time unit). In the case of memory elements, for instance, this is obtained writing a known pattern in the device, exposing it to radiation, and then reading it back to detect mismatches.

Combinatorial circuits are not immune to radiation as Single Event Transients may be generated. Working frequency has a major impact on SETs capture: the higher the frequency the larger the probability of having a memory element corrupted by a propagating transient [Dod04][Eat04]. It is then fundamental for tests to be performed at the operating frequency, to avoid SET underestimation.

When performing radiation tests on complex devices many different resources may be affected by errors, but these do not necessarily appear at the output. In the case of microprocessors, for instance, checking user memory after a test program execution may not be sufficient to characterize its sensitivity. Thoroughly testing a microprocessor under radiation is an expensive and time-consuming task: it would be very attractive to understand the sensitivity of each resource, which errors affect the device computations and which ones are masked, to extend the results collected during the radiation tests to other conditions. These data may permit to predict a device sensitivity as well as a program failure rate. On the other hand, knowing which resources are more likely to fail and how errors propagate gives indication on hardware/software designing rules for lowering device and running program sensitivity.

Accelerated radiation tests are performed using radioactive sources or facilities that accelerate heavy ions or produce neutron beams. Different constraints may be imposed to the test set-up. For instance, the DUT may have to be placed in a vacuum irradiation chamber and high-speed connections may have to be run for several meters and across flanges, making the test preparation quite challenging (and expensive). It is then very desirable to limit the number of cable connections and the speed of the information exchange between the DUT monitoring circuitry and test equipment (host-PC, for instance).

1.2 Motivation of the Work

In the previous Chapter it is stated that radiation effects are a concern for electronics reliability and dependability not only in the space environment, but also at sea-level. For instance, the occurrence of Soft Errors is due to neutrons originating from the interactions of cosmic rays with the atmosphere and even to alpha-emitting contaminants in the package/solder materials. Hence, radiation testing is becoming an important step in the qualification process, especially in the fields traditionally demanding high product reliability, such as the automotive.

Moreover, the overall hardening and mitigation strategies rely on information extracted during different testing and simulation campaigns. Radiation tests can be applied on dedicated test chips aiming at study in detail the sensitivity of the different IPs. On the other hand, fault injection and fault simulations are used to validate the hardening and mitigation solutions at SoC level.

This manuscript and work focus on radiation tests applied at the SoC level, needed to complete the validation of the hardening and mitigation strategies described above. The proposed approach enables the SoC manufacturer and SoC users to setup in a cost effective way the radiation test experiments. In this scenario, beyond the very high number of transistors in a single chip, the complexity of modern devices derives from the integration of different functional modules which would require specific implementation processes. This factor may affect the susceptibility levels towards SEs measured on chip arrays; countermeasures may then be applied at the cores integration stage in addition to the ones introduced at lower levels of abstraction. Efficient strategies are needed to collect data from Systems-on-Chip during radiation experiments and possibly return precise information about the observed phenomena and the most critical parts. The observation of different sensitivities caused by the specific SoC topology and related power supply distribution, for instance, can then be achieved in a realistic way on the final SoC implementation. Other radiation induced effects such as performance degradation may indeed affect the correct interaction among SoCs modules.

SoC radiation testing involves many problematic aspects that partially match the SoC manufacturing test requirements. Major issues for SoC radiation testing are the accessibility to the core boundaries and the diagnostic information retrieval, the test execution frequency, which has to be as high as possible to catch transients, and the test data transfer speed; furthermore, test equipment is constrained by the features of radiation commodities.

This manuscript proposes some guidelines for supporting a low-cost radiation testing methodology for SoCs. The approach is based on the reuse of the Design for Testability/Diagnosability (DfT/D) features added to SoCs for manufacturing test sakes, and on a suitable laboratory setup for applying the tests and observing the results. The shown flows are therefore applicable to any device equipped with the described test structures and not only to purposely design test-chips. The combination of on-chip additional circuitries, a suitable test board and ad-hoc software procedures demonstrates the feasibility and effectiveness of the proposed strategy. As a case study, we will describe the test structures and techniques implemented on a SoC manufactured in a 90 nm technology.

1.3 Thesis Outline

The rest of the manuscript will describe in details the DfT testing structure we applied to SoC radiation tests and the results obtained during several radiation experiments. Some hardening techniques are also taken into account, analyzing their costs and effectiveness. The next chapters are organized as follows:

Chapter 2 – Test Structures: This Chapter proposes an efficient low-cost strategy for collecting data during radiation experiments on Systems-on-Chips (SoCs), exploiting the available on-chip Design for Testability (DfT) structures devised for manufacturing test. The approach combines hardware test and diagnostic features with suitable software tools, which enable accurate measurements and quick transient effects data collection. Specific flows for radiation testing of different kinds of embedded cores are described.

Chapter 3 – The Case Study: This Chapter contains a detailed description of the SoC developed by STMicroelectronics and we tested under radiation. It includes a 64 k bytes memory core, a 16x16 c6288 multiplier as a logic core, and an 8051 microprocessor. Testing board is also described as well as some test control circuitry that vary the DUT working frequency and supply voltages.

Chapter 4 – Embedded SRAM Radiation Tests: After a brief introduction on Radiation Effect in SRAM and how memory are typically tested and characterized, this Chapter described the proposed SRAM radiation test flow that takes advantage of the testing structure described in the previous chapters. Then, experimental setup is presented as well as the radiation test results.

Chapter 5 – Embedded Microprocessor Radiation Tests: This Chapter presents the results of Alpha Single Event Upsets tests of an embedded 8051 microprocessor. Cross sections for the different memory resources (i.e., internal registers, code RAM, and user memory) are reported as well as the error rate for different codes implemented as test benchmarks. Test results are then discussed to find the contribution of each available resource to the overall device error rate.

Chapter 6 – DFM Library Optimization Impact on Alpha Sensitivity: This Chapter presents and discusses the results of Alpha Single Event Upset (SEU) tests on an embedded 8051 microprocessor core implemented in three different cell libraries. Each standard cell library is based on a different Design For Manufacturability (DFM) optimization strategy; our goal is to understand how these strategies may affect the device sensitivity to alpha-induced Soft Errors. The three implementations are tested exploiting advanced Design for Testability (DfT) methodologies and radiation experiments results are compared.

Chapter 7 – TMR Effectiveness to Mitigate Errors Accumulation: To understand the effectiveness of TMR to enhance the device reliability to radiation, we analyzed the alpha induced soft errors rate of circuits hardened with different TMR strategies implemented in SRAM based FPGAs. We first assess the relative sensitivity of the configuration memory bits controlling the different resources in the FPGA. We then study how SEU accumulation in the configuration memory impacts on the reliability of unhardened and hardened-by-design circuits. We analyze different hardening solutions comprising the use of a single voter, multiple voters, and feedback voters implemented with a commercial tool. Finally, we present an analytical model to predict the failure rate as function of the number of bit-flips in the configuration memory to be applied to also to generic devices.

Chapter 7 – Conclusions and Future Works: Conclusions regarding the obtained results and work are drawn. An idea on the future steps to be performed is also presented, regarding high-altitude radiation tests as well as physical-level simulations of the DFM optimized cells.

Chapter 2

Test Structures

Today's highly-integrated System-on-Chips (SoCs) may be composed of up to hundreds of functional modules (cores) of different nature. Their test is a major challenge for the industries as well as for the research community: the reduced accessibility of cores and the physical constraints to be respected make the test procedures more and more difficult with the technology evolution. To enable exhaustive testing while minimizing application costs, Design for Testability (DfT) techniques are heavily employed. These techniques rely on test-devoted hardware integrated on-chip, and include scan chains, Infrastructure-IP and Built-In Self-Test (BIST) modules, suitable wrappers and interfaces. In the early production phases of a new device or technology, the most information has to be extracted from the devices under test for characterization and yield ramp-up. More refined integrated circuitry and techniques may then be employed in the phase that is usually defined manufacturing test.

This Chapter describes in details the testing structures typically used for manufacturing tests of memory, logic, and microprocessor integrated cores. Then an efficient low-cost strategy for collecting data during radiation experiments on Systems-on-Chips is proposed, exploiting the available on-chip Design for Testability structures *devised for manufacturing test. The approach combines hardware test and diagnostic features with suitable software tools, which enable accurate measurements and quick transient effects data collection. Specific flows for radiation testing of different kinds of embedded cores are then described.*

2.1 DfT for Manufacturing Test

In the semiconductor industry, the manufacturing phase deals with the increase of the yield for devices realized in an emerging technology. During the manufacturing phase a large amount of information about the failures affecting the build devices is retrieved and used to characterize the inspected technology by defining its capabilities and constructing limitations, define a set of Design-for-Manufacturing rules suitable to increase the technology quality as soon as possible, and tune the industrial process in order to avoid recurrent constructive defects. Typically, the product yield is very low when starting the development of a new technology, both if using a scaling factor higher than the consolidated one or adopting a different device organization, and it slowly grows until an acceptable quality level has been reached. The fastest is this growth, the shortest is the time-to-market. Fast innovation in VLSI technology makes possible to integrate a complete system into a single chip. In order to handle the resulting design complexity, reusable cores are being used in many SoC applications. Core based SoCs have important advantages as the decrease of the cost of the end-product, and a reduced time- to-market thanks to design re-use.

The manufacturing test of such systems is a major challenge for industries as well as for the research community. Each core embedded into the SoC asks for an accurate test procedure, allowing the extraction of those information required to deeply investigate the causes of technology weakness. Moreover, a quick and cheap overall SoC test plan have to be defined and the description of such test plan have to be easily produced in a convenient language in order to be read and executed by the selected Automatic Test Equipment (ATE). In general, powerful core layer test circuitry are connected using particular bus structures suitably thought to fit investigated SoC, but rather reusable in a different design and, as we will see, in different reliability test, as radiation experiments.

It is clearly stated that better core test quality is achieved by exploiting at-speed executions. In the last decade, this need reflected in the design of several Self-Test architectures. These structures autonomously apply a test sequence, then providing binary information about the test result.

The next paragraphs of this manuscript describe a set of flexible Infrastructure IPs (I-IPs) aiming at the advanced test and diagnosis of memory, user defined logic, and processor cores. This study guarantees the highest possible diversity of involved manufacturing library components and performance parameters.

2.1.1 Embedded memory cores test

Extensive research on fault detection in embedded memories has been performed and various efficient algorithms have been proposed and implemented [Cho97][Mar99] [Iye02][Zor02].

Embedded memory cores often determine the yield in production processes of SoCs, as they are among the ones with highest integration density and tend to consume most of the transistors in SoCs [Coc94]. BIST-based solutions, that provides an effective way to autonomously and automatically generate test sequences, compressing the outputs and evaluating the integrity of memories, are now very popular [Het99][Hua99]. The typical memory BIST implements a March algorithm [Van98], composed of a sequence of March elements, each corresponding to a series of read/write operations on the whole memory or particular locations.

Different approaches have been proposed in the literature to implement March tests: the **hardwired BIST**, the **soft BIST**, and the **programmable BIST**.

The **hardwired BIST** approach is the most widely used. It consists in adding a custom circuitry to each core, implementing a suitable BIST algorithm [Tre93]. The main advantage of this approach is that the test application time is short and the area overhead is relatively small. Hardwired BIST is also a good way to protect the intellectual property contained in the core: the memory core provider needs only to deliver the BIST activation and response commands for testing the core without disclosing its internal design. This approach unfortunately provides very low flexibility as any modification to the test algorithm requires a BIST circuitry redesign.

The **soft BIST** [Tsa01] is a more flexible testing strategy that takes advantage of an on-chip available processor for running a test program. The test program executed by the processor applies test patterns to each core under test and checks for the results. The test program is stored in memory locations containing also the test patterns. This approach uses the system bus for applying test patterns and reading test responses, and it guarantees a very low area overhead, limited to the chip-level test infrastructure. The disadvantage of this approach is mainly related to the strict dependence of the test program on the available processor. As a result, the core vendor needs to develop for the same core different test programs, one for each processor family, thus increasing the test development costs. Moreover, intellectual property is not well protected, as the core vendor supplies to the user the test program for the core under test. Finally, this approach can be applied only to cores directly connected to the system bus and the approach cannot be applied if the core is not completely controllable and observable.

An alternative approach is the **programmable BIST** [Dre98][App03]. The core vendor develops a DfT logic, which wraps the core under test and includes a small custom processor, which is exclusively devoted to test the memory. There are various advantages in the use of this testing architecture. The intellectual property, for instance, can be protected as only one test program has to be developed and the design cost for the test is then vey reduced. This technique provides high flexibility since any modification of the algorithm simply requires a change in the test program. Thanks to the efficiency of the custom test processor the test application time can be taken under control and the test can consequently be executed at-speed. Finally, each core is autonomous even from the test point of view as the core test simply requires the activation of the test procedure and results reading. The main potential disadvantage is the area overhead introduced by replicating the custom processor in each core to test. However, due to the very limited size of the processor with respect to medium and large size memory cores area, this problem is marginal and may also be overcome when sharing the BIST circuitry among many memory cores.

Eventually, information about failures, in terms of fault location and type, can be collected resorting to more complex flows and algorithms (soft BIST approach) and/or to additional hardware devices (hardwired and programmable BIST), such as registers storing results.

In our analysis, as described in paragraph 2.2.1, we decided to adopt the programmable BIST (pBIST) approach for the radiation experiments on the SRAM core. The pBIST applies a selected user-defined March tests and is based on the definition of a custom instruction set. The test sequence is memorized in a code buffer, fetched and decoded by a suitable control unit, and finally physically applied to the embedded memory core. This feature allows simplifying the customization of the memory test without any re-design cost and enables further diagnosis inspection as extensively documented.

The proposed architecture and the defined result analysis flow allow building the faulty memory bitmap as well as the complete download of March execution results which, as described in the following paragraphs, is extremely important for radiation tests sake. Moreover, this process is performed at-speed and avoids the aliasing introduced in compression strategies.

2.1.2 Embedded logic cores test

Concerning logic cores, the available approaches are usually grouped in the following classes: **scan-based**, **synergy-based**, and **BIST based**. In the Scan-based and logic BIST approaches, a set of test patterns are generated using Automatic Test Pattern Generation (ATPG) and applied to the circuit. In the sequential approach, the calculated patterns are sequentially sent to the circuit and responses read after each application and any additional internal structure is added in order to improve the effectiveness of the patterns. On the contrary, in the scan approach, the controllability and the observability of the circuit are improved by modifying the common flip-flop: the scan cells allows writing and reading the content of the memory element during the test application, and are connected to compose a scan chain. However, as a serial process is required to load and upload the scan chain, this approach requires onerous application time and heavy ATE requirements in terms of storage needed for test data and test application program.

For our analysis we adopted an alternative and well documented technique, based on pseudo-random pattern generation. Such approach is based on Galois theories for the generation of pseudo-random number sequences starting from the definition of a characteristic polonium. Particular structures, called Autonomous Linear Feedback Shift Registers (ALFSRs), implement such kind of pattern generation strategy [Str02][Bar87].

2.1.3 Embedded microprocessor cores test

Almost every modern SoC includes at least one microprocessor or microcontroller core, which may be a general or a special purpose processor, surrounded by different memory cores of various size used for code and data storage. Unfortunately, the complexity of SoCs including deeply embedded cores often makes their testing very hard.

Self-test techniques are expected to play a key role into this scenario [Tsa01].

Self-test approaches can guarantee high fault coverage by autonomously performing the test at the nominal frequency of the IC and drastically reduce the cost of the required Automatic Test Equipment. There are two different categories of self-test approaches: **Hardware-based Self-test** and **Software-based Self-test**.

Hardware-based Self-test architectures require additional hardware structures (e.g., Built-in Self-test, Logic Built-in Self-test, etc.). The adoption of these techniques is particularly suited to IP cores not strictly constrained in terms of timing and consumption [Het99] as hardware modifications to support Self-test allow at-speed test and relieve the ATE from test management. However, whereas processor cores are performance-constrained, any introduced additional logic can fatally impact their efficiency and power consumption.

Software-based Self-test (SBST) methodologies appear to better suit embedded processor cores test. Software-based strategies are based on the execution of suitably generated test programs [Bel82][Tha80]. Therefore no extra hardware is required, and the existing processor functionalities are used for its test. No modification of the IP are needed and the performance is not decreased, since the test is performed at the operative speed for the embedded core. Several efforts were made to devise effective techniques for generating test programs able to obtain high fault coverage figures at acceptable costs and, recently, some significant results were achieved in this field even when pipelined processors are considered [Kra03][Cor03]. To realize this strategy, a memory module for storing the test program, a mechanism to upload the code, and a method to start the self-test program execution should be identified. Finally, a procedure to monitor the program execution and extract test results should be defined. Software-Based Self-Testing is definitely an increasingly valued methodology. It provides an affordable solution relying on the execution of suitable self-test programs, utilizing the processor ISA instructions in normal mode of operation and not requiring core design modifications. Additional diagnostic benefits are introduced with the application of test program execution interleaved with scan operations.

The Software-base Self-test seems to meet most of the requirement of our tests, and in paragraphs 2.2.3 a detailed description of the SBST used for the microprocessor core radiation test is given.

2.2 Proposed Strategy for Radiation Tests

The proposed strategy for radiation testing addresses SoCs including different kinds of cores, and aims at characterizing the sensitivity of the different modules to radiation, considered both separately and in their interaction.

There are three basic ways to find the Soft Error Rate of chips [Zie04]: field analysis, life testing, and accelerated testing. In this manuscript we focus our attention on SoCs accelerated radiation testing. Accelerated tests are performed using various beams of particles that simulate cosmic rays or radiation sources. The device is exposed to an accelerated particles flux, emulating the natural environment. As described in this paragraph, accelerated radiation experiments may impose strict constraints to the test setup, that must be taken into account when designing a test strategy.

Our goal is to test a SoC exposed to radiation and understand the impact of the different cores corruption on the overall system failure rate. Radiation tests can be applied on dedicated test chips aiming at studying in detail the sensitivity of the different Intellectual Property cores (or simply IPs) composing a System-on-Chip. The integration of different functional modules supports the complexity of modern devices and usually requires specific implementation processes. This factor may affect the susceptibility levels towards Soft Errors (SEs) measured on chip arrays.

The selection of a SoC as a case of study is also motivated by the need of observing different sensitivities caused by the specific topologies and related power supply distribution, which can be achieved in a realistic way only by testing the final SoC implementation and may not be observed testing stand-alone cores or cell arrays. Other radiation-induced effects (such as performance degradation) may indeed affect the correct interaction among SoC modules and can be only experimentally observed and measured in a complete system.

This manuscript then focuses on radiation tests applied to microprocessor, logic and memory cores embedded in a SoC and aiming at determining their alpha-induced soft error rates and how their corruption contributes to the overall system failure rate.

Typically, two radiation testing modes are applied to the devices, *static* and *dynamic*: the first performed to identify the susceptibility of state holding elements to bit-flips without stimulating them, and is applied to memory arrays and flip-flops in the logic holding specific logic states; the second also takes into account the effects of SETs on the combinational logic and is applied in operating conditions, i.e., by actively stimulating the circuitry at working frequency.

The methodology is based on the reuse of the techniques and structures devised for manufacturing test and already available on-chip, which are used to access, activate and control deeply embedded cores and internal resources, while reducing the requirements for external test equipment. Test-devoted structures include, on the one side, BISTs and I-IPs that allow applying stimuli to and observing results from the circuitry under test and, on the other side, suitable communication infrastructures to control the test execution and to transmit test data. More in detail, the embedded structures allow to:

- easily reach and set defined states of state-holding elements (memory arrays and flip-flops)
- activate the logic *at-speed* (i.e., at nominal frequency) and beyond, if needed
- ease precise Soft Error diagnostic information retrieval.

Our intention is to built a radiation test setup which is low-cost and easy to use. The general methodology reduces the costs of radiation testing experiments in the following ways:

- avoiding the adoption of ad-hoc test equipments and long high-speed connections between DUT and controlling hardware
- relying on already available SW to manage data collection and classification
- reusing some of the equipment enabling low-cost manufacturing test.

In order to be effectively used for radiation testing, the implemented DfT requires some common general characteristics, which reflect traditional requirements for low-cost test. First of all, the test logic has to be easily resettable to a known state in order to discard any effect of radiation exposure prior to test application and test results downloading. Then, each core needs to be isolated from the other cores during its testing, and a common test interface has to be provided. For these reasons, a chip-level test data transfer infrastructure may consist of suitable interface wrappers and connections, such as the ones proposed by IEEE 1500 [IEE05] and IEEE 1149.1 (JTAG)

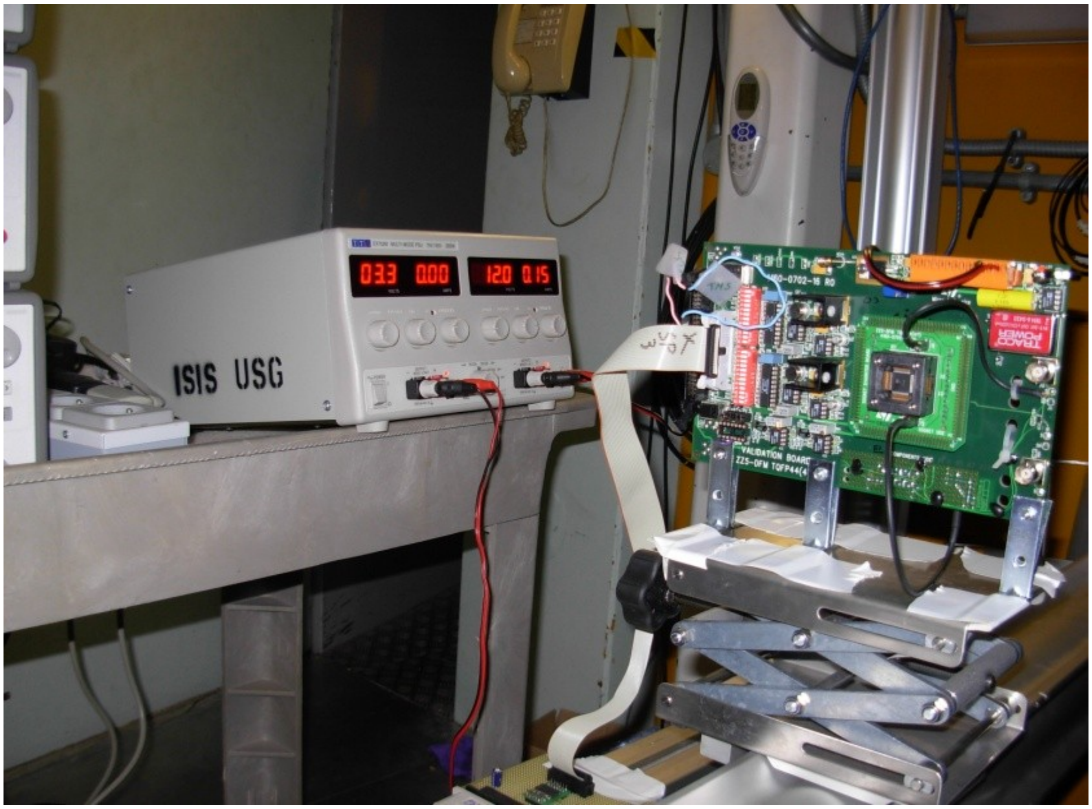

Figure 2.1: *ISIS irradiation room. The Device Under Test should be aligned with the neutron beam, while complex and delicate test support electronics (as host PC or ATE) should be placed outside the irradiation room. Several meters of cables are then needed to connect them.*

[IEE94] standards.

As stated above, there are various constraints that may be applied to the radiation test setup, depending on the facility in which experiments are performed. For instance, in the case of ISIS pulsed neutron source, at the Rutherford Appleton Laboratory in Didcot, UK (Fig. 2.1), the test support hardware (which may be a host PC or dedicated ATE) must be placed several meters away from the Device Under Test (DUT) to avoid control circuitry corruption. In fact, as neutrons are very difficult to shield, testing hardware should be placed outside the irradiation room, where neutron flux is even lower than the natural one. Long connections are then needed between DUT and host PC or ATE. As explained in the previous paragraphs, even in this scenario, logic must be tested at speed to avoid Single Event Transient underestimation. To do so, testing circuitry must control the DUT at working frequency and even beyond, if possible. This makes high speed long cable connections between DUT and control hardware necessary,

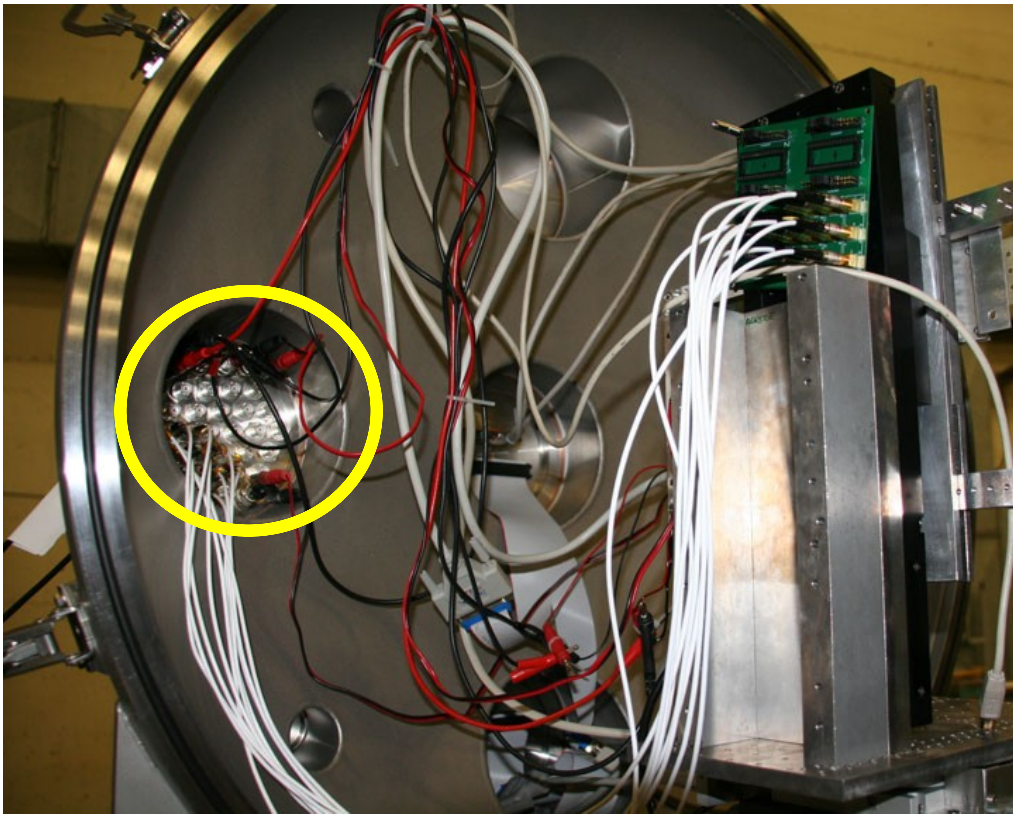

Figure 2.2: *SIRAD irradiation chamber. The Device Under Test should placed inside a vacuum chamber, while complex and delicate test support electronics (as host PC or ATE) should be placed outside it. Only a limited number of feed-through connectors (highlighted in yellow in the picture) are available.*

which may be very expensive. As we will see in paragraph 2.3, our proposed strategy overcome this issue, as JTAG will be the only interface between a host PC or low-cost testing hardware (i.e. a PIC-based board), thus permitting low speed parallel cable connections.

In the case of heavy ion beam tests, the DUT must be usually placed inside a vacuum chamber to avoid accelerated particles interaction with air and permit them to reach the tested device silicon die. Fig. 2.2 shows the SIRAD irradiation chamber, at Laboratori Nazionali di Legnaro, INFN, Padova, Italy. In this case only a limited number of feed-through connectors are available and thus the number of connections between the testing hardware and DUT should be minimized. Once again, the JTAG interface seems to meet the requirement and ease radiation test of the device as relies on only 5 signals.

The following paragraphs give an overview of the structures adopted for the radiation test of the different cores. In each of the proposed cases we start from the DfT structure that seems to be easily applicable to radiation test and try to find the best solution to overcome radiation test constraints and ease error detection without affecting experimental results precision.

2.2.1 Embedded SRAM radiation test flow

Its well known that SRAM may be corrupted by radiation. An impinging particle may indeed have enough energy to reverse the stored bit value, generating a bit-flip or a Single Event Upset. Moreover, with technology shrinking it is possible for one single particle to interact with more transistors, leading to Multiple Bit Upsets. The goal of radiation experiments on embedded SRAM is then the *static* measure of memory arrays susceptibility to radiations while no stimuli are applied and, eventually, a *dynamic* measure to detect radiation induced Multiple Bit Upsets. In this paragraph we explain how DfT structures may be used in SRAM embedded core radiation tests.

In the manufacturing test flow, memory test and diagnosis usually rely on the

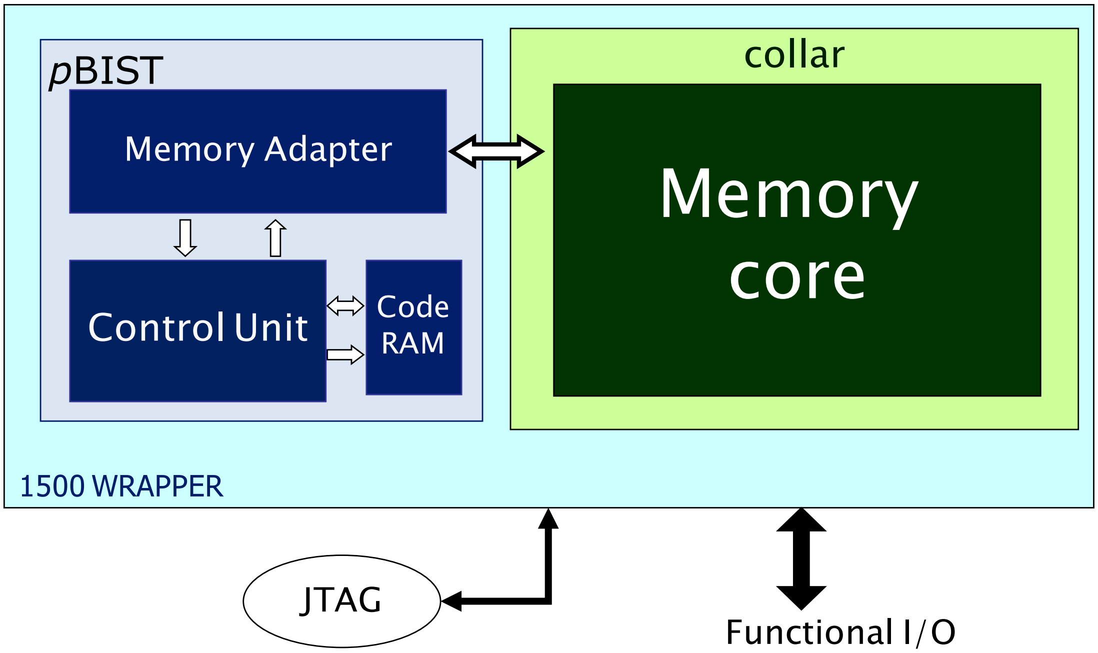

Figure 2.3: *Proposed low-cost embedded memory core radiation test structure, including a programmable BIST composed of a control unit, dedicated code RAM, and memory adapter, IEEE 1500 wrapper, and JTAG interfaces*

execution of a series of diagnostic March algorithms on the memory array, and allow to spot failing locations and determine the type of the discovered faults. For the latter concern, a very useful feature is the programmability of the memory test algorithm to be applied, which is a prerogative of programmable BIST architectures.

In Fig. 2.3 the schematic diagram of the pBIST based test structure for the embedded memory core is represented. This architecture [App03] permits the application of word-oriented tests for memory by loading a small internal **code RAM**, totally dedicated to store the test algorithm. When the test code has been loaded, the code RAM is checked calculating a signature to detect any mismatch. Once the code memory has been properly and correctly loaded, a control unit fetches and decodes the instruction loaded in the code memory during the initialization phase, and finally a memory adapter module applies test vectors to the DUT.

The **Control Unit** manages the test program execution, receiving commands (i.e., START, RESET, etc…) from the external testing hardware, and fetches/decodes the instruction from the code memory. The Control Unit includes an Instruction Register (IR) and a Program Counter (PC) and update some Memory Adapter registers to customize the test and diagnosis procedures.

The **Memory Adapter** includes all the test and diagnosis registers used to customize and correctly execute the March algorithm. Those registers are: the Control Address that contains the address of the currently accessed memory cell; the Control Memory registers that contain the data to be written to the memory or the data read from the memory; and, finally, the Control Test and Result registers.

The flexibility of the programmable diagnostic BIST approach can be fruitfully exploited for both static and dynamic radiation testing of the memory core. In fact, the programmable BIST engine can be used to write specific configurations into a memory to prepare the array for the radiation experiment and, after radiation exposure, for reading out bit-flip positions, without requiring direct write and read operations from the external tester. Memory content set-up and result retrieval is very fast even compared to the soft-BIST methodology. The employed flow for the static test consists of three steps to be iterated when the SoC is exposed to radiation:

1. memory content preparation using the programmable BIST running a simple pseudo-March algorithm, e.g., $\{\uparrow(w0)\}$

2. wait for a defined period and let radiation induced errors to accumulate

without stimulating the device

3. check the entire memory array for mismatches using the programmable BIST running a simple pseudo-March algorithm, e.g., {↑(r0)}, and download through the JTAG low speed connection information about failures

The SRAM is initialized by writing a known pattern on the entire array and, after a predefined time of radiation exposure, scanned to detect mismatches. The process is iterated many times to collect data varying operating parameters such as supply voltage to study device sensitivity dependences.

For the dynamic test of the memory core the test flow is very similar to the static test one. The memory content is prepared and continuously monitored during radiation exposure (skipping static test step 2 and continuously repeating step 3). If the detected errors number increment is greater than one in two consecutive checks, MBU may be occurred.

The only information exchange between the host PC and the BIST regards the pattern to be written (few bits to be loaded in the code RAM module) and errors detected, permitting the use of simple and low-cost connections. The DUT may be exposed to radiation even during memory content preparation and readback procedure as these processes are handled by hardware at high frequency and, as reported in Chapter 4, it is very unlikely for alphas or neutrons to corrupt a bit during their executions.

2.2.2 Embedded logic core radiation test flow

Radiation testing on combinational logic aims at determining its *dynamic* sensitivity to Single Event Transients. The number of errors caused by SETs strongly depends on frequency [Dod04], it is thus fundamental to perform tests at DUT operating frequency. Moreover, as the incidence of SETs is expected to increase with frequency, while the one of SEUs is not dependent on that, to distinguish the contribution of SEUs and SETs, experiments have to be performed at different frequencies.

As tests must be performed at high speed to avoid SET underestimation, stimuli must be applied at the Device Under Test (a 16x16 bit c6288 multiplier in our case) working frequency and beyond, if possible. As explained in paragraph 2.2, some radiation test facilities impose long connections between DUT and controlling hardware, and high speed long connections are usually very expensive. Thanks to the JTAG interface, the logic BIST may be initialized and results downloaded at low speed, but yet tests will be performed at high speed by the DUT integrated testing structures.

As described in paragraph 2.1.2 and showed in Fig 2.4, a commonly employed strategy for the manufacturing test of embedded combinational cores exploits BIST for generating pseudorandom patterns through ALFSR and compressing the results through Multiple Input Shift Register (MISR) modules. The **Control Unit** manages the test execution by receiving and decoding commands from the control signals. It receives the number of patterns to be applied to the logic core, drives the test enable signal that starts and stops the test execution and selects the result to be uploaded. The multiplier inputs are given by the **ALFSR** [Str02][Ber87] pseudorandom generated patterns and by a **RETRO** register, inserted to add controllability and observability in the diagnosis process, as its content can be programmed and read from the outside. ALFSR pseudorandom patterns are generated starting from a 16 bit seed sent to the logic BIST through the JTAG and wrapper interface. The logic core is then stimulated at high speed for a programmable number of steps while MISR circuitry monitors executions computing a test signature.

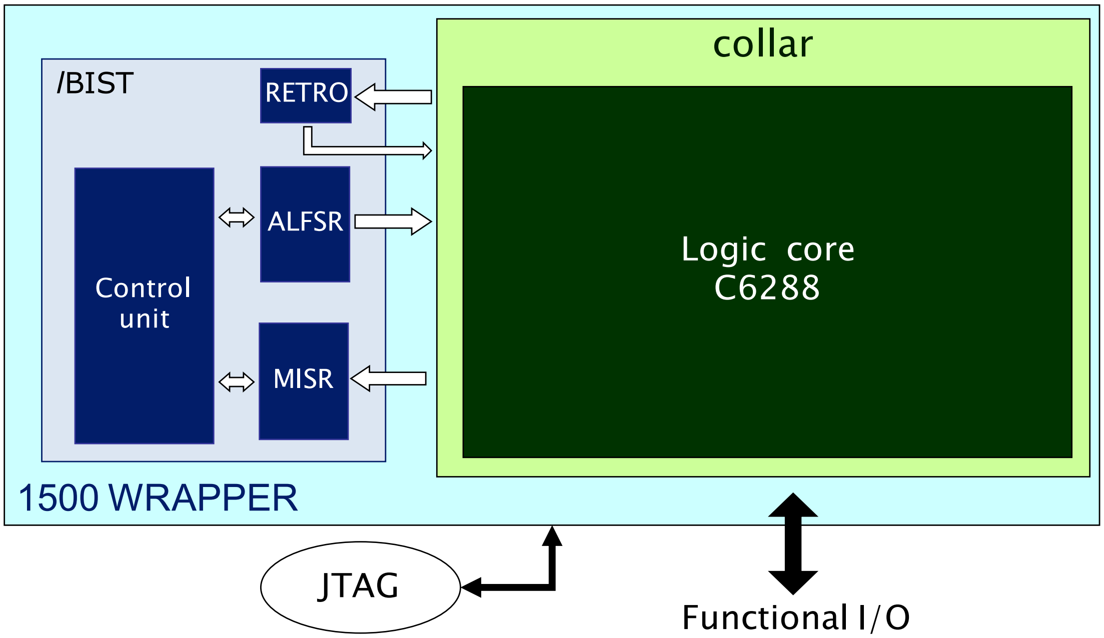

Figure 2.4: *Proposed low-cost embedded logic core radiation test structure, including a logic BIST composed of a control unit, an ALFSR-based pseudorandom patterns generator, and a MISR to detect errors, IEEE 1500 wrapper, and JTAG interfaces*

This flow suits well for radiation testing. Once the pseudorandom generator has been programmed with the chosen seed and the number of steps has been loaded through the JTAG at low speed, test patterns are autonomously applied at high frequency by the BIST circuitries. While the device is exposed to radiation, tests can be performed at high frequency without the need of an expensive ATE to monitor the test flow. Parameters setting up and compressed results downloading are performed at low frequency, again permitting the use of low-cost connections.

2.2.3 Embedded microprocessor core radiation test flow

In a processor core both state-holding elements and combinational logic are present. Radiation testing aims at determining the sensitivity of flip-flops (static test) and to verify the overall behavior of the circuit under radiation (dynamic test).

As described in paragraph 2.1.3, processor manufacturing test and diagnosis may be based on the execution of suitably selected SBST procedures, which encompass the following operations:

- upload the test program in a suitable memory area
- activate its execution, i.e., letting the program run, stimulating the components
- retrieving the results opportunely stored during the test program execution

The structure of the SBST we design and develop is described in Fig. 2.5 and exploit the internal execution of a suitably generated test program. The execution of such a test program, loaded from the outside into a selected memory space through the JTAG low speed interface, allows stimulating and observing many parts of the processor core and is launched exploiting processors features, like internal interrupt ports. It is important to notice that with the proposed strategy the test are performed at the same working frequency of the processor itself without any modification on its internal structure. Test code upload, start and result collection are performed by an I-IP [Jac93], which includes test control circuitry and MISR that compresses the microprocessor output port values during codes executions.

The test code is loaded into dedicated memory location reusing the system bus, so the I-IP is connected directly to the memory and then takes control of the bus acting on the processor functionalities by means of driving its address, data and control port to high-impedance or running special procedures to move data to the memory. The self-test procedure is activated taking advantage of the interrupt mechanism supported by the processor, transforming the self-test code into an interrupt subroutine. The address of the uploaded self-test code is stored in the slot of the Interrupt Vector Table corresponding to the interrupt triggered as soon as the controlling hardware sends the activation command. The interrupt signals are managed by the wrapper circuitry (see paragraph 2.3) in charge of activating the self-test procedure. During test execution, the code memory is continuously monitored so to distinguish between code ram errors and execution errors. To ease results monitoring, some instructions are needed in the test program to transfer the fault effects to some easily accessible observability point (i.e., external ports). To release the testing hardware from monitoring the test execution, which must be performed at high frequency, a 32 bit wide MISR computing a test signature from the output port values and storing the final test has been added. MISR characteristics are tuned to ensure a sufficiently low percentage of aliasing and an acceptable silicon area overhead.

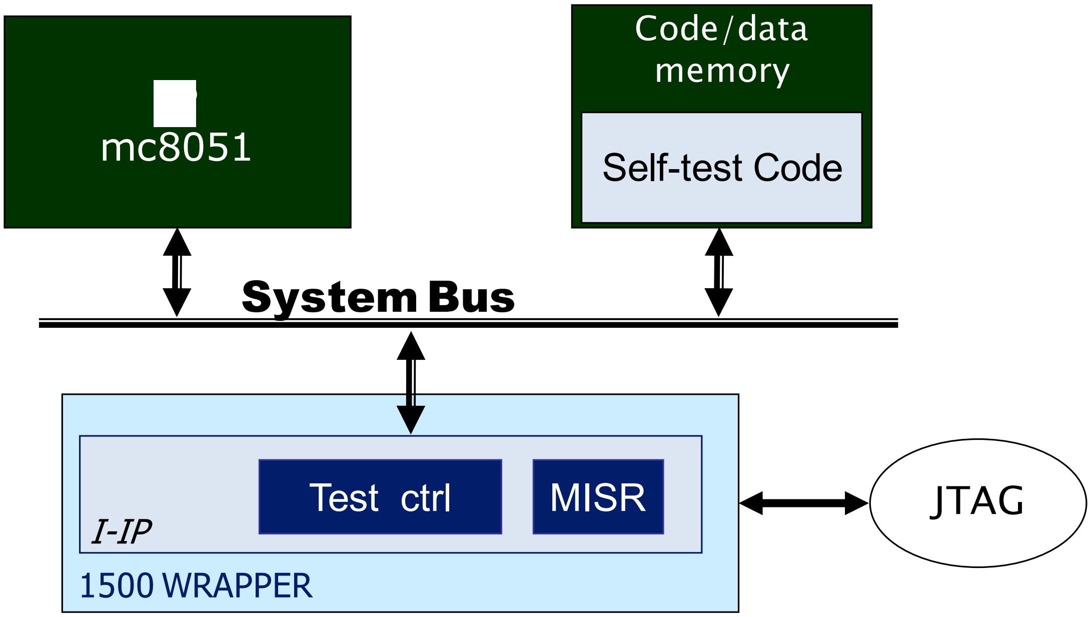

Figure 2.5: *Proposed low-cost embedded microprocessor core radiation test structure, including a Software Based Self Test composed of an I-IP with a control unit and MISR to detect output errors, IEEE 1500 wrapper, and JTAG interfaces*

Finally, a watchdog has been added so to monitor code executions and detect any wrong program flow or microprocessor halting condition.

This test structure is particularly easy to use under radiation, and permits to gain important information about the radiation induced effects on the device when a code is being executed.

One again, under dynamic condition is very important to perform tests at high frequency to avoid SETs underestimation. JTAG interface permits the use of low-speed and thus low-cost connections from the DUT to the controlling hardware or host-PC but tests are performed at high frequency by the built-in test structures.

The following steps are performed to characterize DUT sensitivity to radiation:

1. **Static test:** the static test of a microprocessor aims at measuring its memory elements radiation sensitivity. It is similar to SRAM radiation test flow, but employing scan chains for loading and reading the flip-flop contents. All the accessible memory location are reset (or set) shifting 0s (or 1s) in the scan chains. Eventually, an ad-hoc procedure that reset (or set) all the accessible registers may be loaded in the code memory and then executed by the SBST. After a predefined period of time during which the device is exposed to radiation without being stimulated, we check for mismatches. This may be done shifting the scan chains or executing an error checking code that reads memory location contents, detects radiation induced bit flips number and report it to output port.

2. **Dynamic test:** the dynamic test of a microprocessor aims at measuring its radiation sensitivity under operating conditions when a particular application is being executed. The dynamic test may be performed with the following steps:

   a. upload the test program in a suitable memory area, opportunely protected against radiation

   b. expose DUT to radiation

   c. launch the test program

   d. wait for a specified number of clock cycles

   e. stop the processor clock

f. remove DUT from radiation exposition

g. download scan chains content

h. restore correct contents if test program functionality is compromised

i. restart from step b

During static tests, reused DFT does not impact on the results since download operation is performed at low frequency. Therefore errors due to SETs in the scan chain combinational logic are unlikely to appear. SEUs affecting flip-flops during scan operations contribute to the SEU sensitivity measures. Concerning dynamic tests, DfT is not active during exposition times.

2.3 Test Interfaces

The different cores and the suitable testing strategies described in the previous paragraphs have to be integrated in one single chip. Our intention is to test the different cores under radiation both isolated and in their interactions. To do so we need to develop efficient interfaces to easily access the test structures and eventually isolate the core under test. Moreover, as mentioned in the previous paragraphs, it is very useful to have low-speed connection between the host PC or controlling hardware and the DUT and a limited number of wires is also desirable. The widely used IEEE 1149.1 (known also as JTAG) [IEE94] interface meets these characteristics and so, as stated in the previous and in this paragraphs, we decide to apply the JTAG interface to our testing strategy. Communications between DUT and testing hardware then rely on only 5 signals.

To provide a common interface to the test logic we added IEEE 1500 wrappers [IEE05] for each core. This standard defines test interface architectures, called wrappers, which allow, besides flexibility and easy reuse, the usage of high-level description in Core Test Language (CTL) [Mar99]. A general schematic structure of an IEEE 1500 standard compliant wrapper is shown in Fig. 2.6. Such test interfaces are connected by a test bus that constitutes the Test Access Mechanism (TAM) allowing reaching every core embedded in the SoC. Every core integrated in the SoC share a test access method based on the IEEE 1500 standard and its designed testing I-IP can then be controlled through a common access protocol. The JTAG Test Access Port (TAP) controller addresses the Wrapper Instruction Register (WIR) of the wrapper surrounded the desired core and serially load instructions and/or data to perform the core test or download results. The characteristic of registers and wrapper depends on the surrounded core. In addition to the mandatory IEEE 1500 standard components, we added to each core some registers to send instructions and data to the testing I-IP or read test results:

- Wrapper Control Data Register (WBCD) through which the TAP controller sends the commands to the testing I-IP
- Wrapper Data Register (WDR) which is an I/O buffer register. The TAP controller can read the diagnostic information stored in the result registers of the addressed testing I-IP or write data used by the testing I-IP

The external test controller can then interact with the on-chip devices by sending high-level instructions to the TAP controller using just a low-speed connection relying on 5 signals. Some details about testing structure physical implementation and characteristics are given on the following chapters.

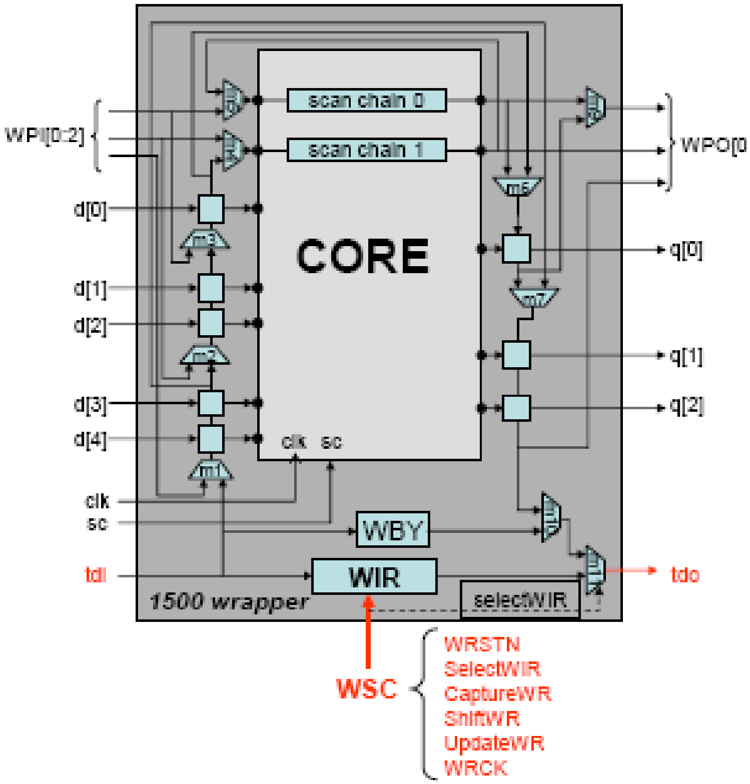

Figure 2.6: *a generic structure of an IEEE 1500 standard wrapper [IEE05]*

Fig. 2.7 shows the schematic view of a generic core surrounded by IEEE wrappers with WBCD and WDR registers. It's important to notice how while the circuit test clock (CLK in the picture) run at high frequency, the testing structures are initialized at low speed using a separated clock line (WCLK in the picture). The testing structures are controlled and test results are downloaded through the JTAG chain that connects all the cores' wrappers.

In the following Chapter, the resulting case study SoC is presented. An SRAM core, a logic core, and a microprocessor core are integrated in a single SoC. Specific test structures are applied to the different cores, as described in this Chapter, and each structure has been designed and implemented to ease the test of the target core. IEEE 1500 wrappers surround all the available cores and the JTAG is the only interface

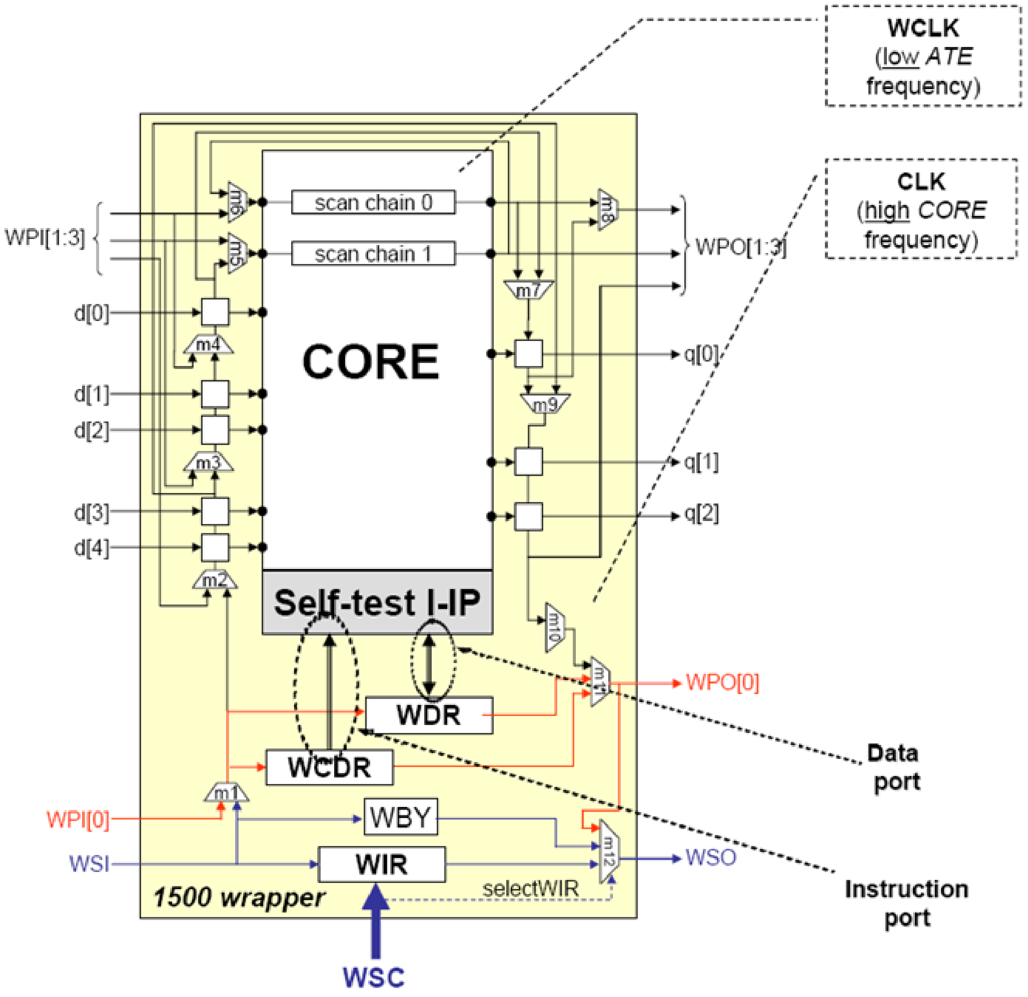

Figure 2.7: *the structure of an IEEE 1500 standard wrapper in which WBCD and WDR registers are added*

between the SoC and a host PC or controlling hardware, thus permitting just a low-speed parallel cable connection to the DUT.

2.4 Conclusions

Modern SoCs include various different interacting cores. Their different natures and characteristic impose different testing structures, each devoted to the test a specific core. Moreover, the cores are deeply integrated in the SoC, and thus efficient strategies are also needed to access the core and extract information about test executions. Design for Testability strategies are very effective and widely used in the manufacturing test of different devices.

In the case of radiation experiments there may be some constraints to the test setup that make even harder the test executions. The main issue when performing accelerated radiation test regards the connections between DUT and the controlling hardware, which may be long and limited in number.

We propose the reuse of some DfT structure for radiation test sakes. The built-in structures permit to test the cores at operating frequency, and to achieve precise radiation induced errors information. Thanks to the JTAG interface, just low-speed connections are needed between the host PC and the DUT, and wrappers allow an ease access to the different cores dedicated testing structures.

Chapter 3

The Case Study

In the previous Chapter we described the commonly employed test strategies for integrated cores. An overview of Design for Testability circuitry for a memory core, a logic core, and a microprocessor core test is also given and a strategy to be applied to radiation tests is proposed.

The advantages of the reusing the DfT testing circuitry are manifold. As we have seen, just low speed connection are needed between the Device Under Test and the host PC or controlling hardware, and, as we adopt the JTAG interface, communications rely on just 5 signals. Those characteristics fit well with most of the accelerated radiation test facilities setup constraints. We also stated that to have a reasonable characterization of the radiation effects in the cores or in the overall system, tests should be performed at high frequency. DfT integrated testing structures, indeed, stimulate the cores and monitor their executions at the same DUT working frequency, thus avoiding Single Event Transients underestimation.

This Chapter contains a detailed description of the SoC developed by STMicroelectronics that we have tested under radiation. It includes a 64 k bytes memory core, a 16x16 c6288 multiplier as a logic core, and an 8051 microprocessor. Testing *board is also described as well as some test control circuitry that vary the DUT working frequency and supply voltages.*

3.1 The System on Chip Architecture

The test strategy proposed in Chapter 2 has been realized on a test vehicle manufactured by ST microelectronics in a 90 nm CMOS technology. The developed SoC includes a 64Kx8 bit sized SRAM memory (built with perfectly symmetric and balanced bit cells), a 16x16 parallel multiplier, and an 8-bit microprocessor. As described in detail in the previous Chapter, it is possible to achieve high diagnosability for each of the cited components resorting to the following test structures:

- a March-based programmable diagnostic BIST (pBIST) used for memory core test [App03]
- a parametric logic BIST (lBIST) the multiplier is equipped with [Ber05_1]
- an Infrastructure-IP (I-IP), which manages the execution of SBST procedures on the processor [Ber05_2]
- additional scan structures inserted for the sake of observability and controllability of the final test, and for comparison with traditional test/diagnosis flows.

IEEE 1500 wrappers are provided for each core in order to provide a common interface to the test logic. The external test controller interacts with the on-chip devices by sending high-level instructions. Finally, the test structures are accessed through IEEE 1149.1 (JTAG) TAP controller.

The conceptual view of the overall system on chip is given in Fig. 3.1. Besides the three core to be tested, the dedicated testing structures and interfaces are represented. Thanks to the proposed structure, we are able to characterize both the single core and the overall system radiation sensitivity. In this System on Chip the different cores are able interact, as they share the same bus and the microprocessor core is connected both to the multiplier and to the SRAM core. Part of the SRAM core, in particular, is used as Code RAM, User RAM and Data RAM by the microprocessor.

The resulting System on Chip includes cores that are typically used in modern devices. The different nature of the cores permits us to evaluate our testing strategy both for memory, logic elements and when the two interact.

Prior to silicon implementation, the overall System on Chip design was described in VHDL so to evaluate area overhead and the efficiency of the proposed architecture. As we will see in the following chapters, VHDL description is also very useful as it permits to simulate radiation induced errors effects and propagation.

As a first characterization, in order to evaluate the effectiveness of the approach in terms of hardware cost, the RT-level behavioral VHDL description of the I-IPs and their Wrapper have been synthesized using the Synopsys Design Compiler tool with a generic gate library.

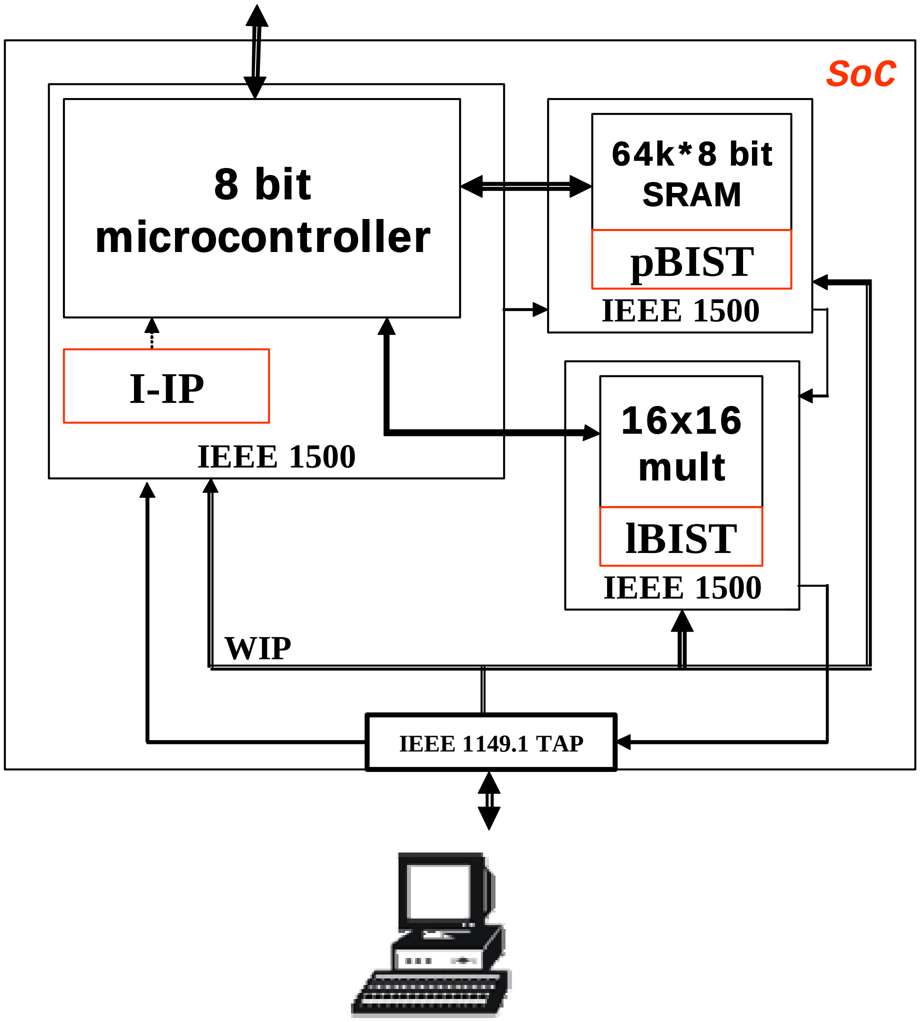

Figure 3.1: *The case study System on Chip schematic view*

The SRAM embedded core is composed of 64 k x 8 bits manufactured by STMicroelectronics using a mixed/power 90 nm CMOS technology. The pBIST structure described in Chapter 2 is composed of a small processor named Control Unit (760 equivalent gates) with a 256 bit code memory, a Memory Adapter module (4,027 equivalent gates) and the Wrapper (2,944 equivalent gates). The overall pBIST area is then less than 1% of the DUT one.

The logic core is a 16 x 16 c6288 multiplier. The lBIST has to generate pseudorandom patterns starting from a 16 bit seed and RETRO register value for a predefined number of steps. The overall pBIST structure, including Wrapper, is composed of 15,837 equivalent gates.

The microprocessor is an Intel 8051. It uses two internal memories: a 64k byte size ROM memory and a 256 byte size RAM memory for registers, stack and variables. Programs codes are stored in the SRAM core connected to the microprocessor parallel ports. In the Intel 8051 case, the silicon area overhead due to the test structures is almost entirely due to the introduction of the 1500 wrapper, while the introduction of the I-IP as a support to the self-test approach (which simply activate the tests triggering the interrupt and monitors execution through the MISR circuitry) results in less than 2% of the additional area. In fact, the processor core is composed of 37,417 equivalent gates, while the testing I-IP of just 490 equivalent gates and 1500 Wrapper of 1,580.

The overall System on Chip area overhead due to the testing structure is shown in table 3.1. As reported, the overhead introduced by the built-in test architectures is very small (2.57 %) with respect to the chip size. Moreover, as demonstrated in Chapter 2, the cores efficiency in terms of working frequency is not affected as the test structures were design with the constraint of maintaining the cores performances unaltered.

Table 3.1: *Area overhead at gate level for DfT*

Core	**Additional structure**	**Size [#gates]**
µP	I-IP	12,159
SRAM	pBIST	9,068
16x16 Multiplier	lBIST	15,837
	Original SoC size	1,442,137
	Overhead	***2,57 %***

3.2 The System on Chip Physical Implementation

STMicroelectronics implemented various chips with the proposed architecture. The main objectives of the collaboration is to built a system which is easy to test, and to realize a System on Chip that could be heavily applicable in complex automotive designs. The automotive application require both high reliability, high robustness and low power consumption. The proposed DfT testing structures will allow us to deeply characterize the device to detect faults or weak components as well as in the field executions monitoring.

In Fig. 3.2 a picture of the silicon die in which the SoC has been implement is reported. It’s easy to distinguish between the microprocessor core (on the left in the picture) and the SRAM core, on the right. The SRAM structure is regular, as area consumption is the main concern when building a memory core, while in the case of the microprocessor, speed must be optimized. The small c6288 core is inserted between the

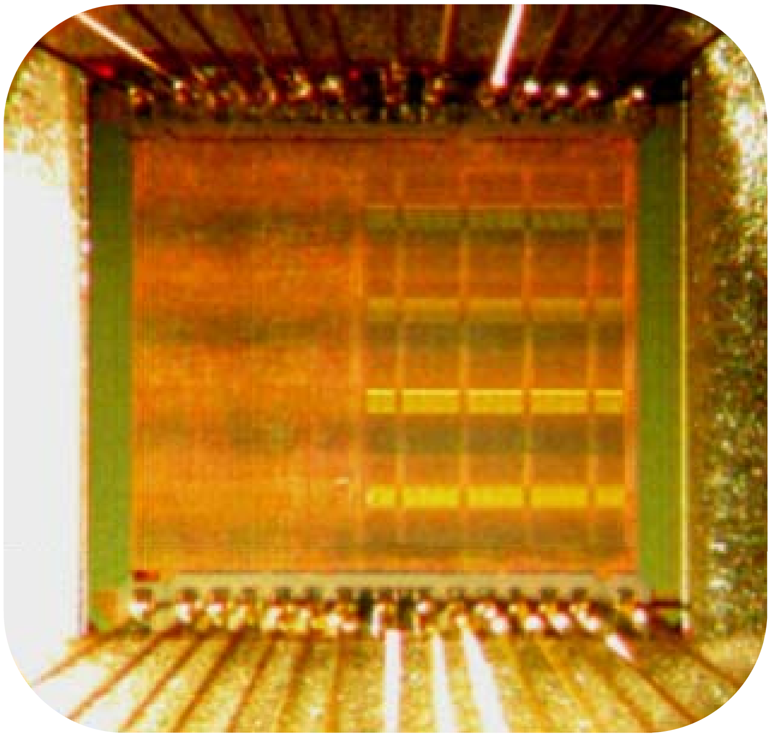

Figure 3.2: *Picture of the case study System on Chip implemented by STMicroelectronics. On the left side there is the 8051 core and on the right the SRAM core, with a regular structure, and between them the small c6288 multiplier.*

microprocessor and the SRAM, in the picture is just a thin strip in the middle of the silicon die.

STMicroelectronics build the devices in two different packages, the plastic and the ceramic. The plastic packaged devices are the ones to be widely produced, and have been used to perform DfT tests and measure the robustness of the chip using the integrated structures. On the contrary, the ceramic package devices were built for radiation tests purposes. In fact, for some radiation experiments sakes it is fundamental to have the silicon die of the device completely exposed to radiation to permit the impinging particles to interact with its active area. Americium produced alpha, or heavy ion accelerated at the SIRAD facility, in Legnaro, Italy, for instance, have a very short penetration range and even few µm of plastic may stop them. As the ceramic package of the System on Chip can be easily removed, the active area is completely exposed, thus again easing radiation test of the device.

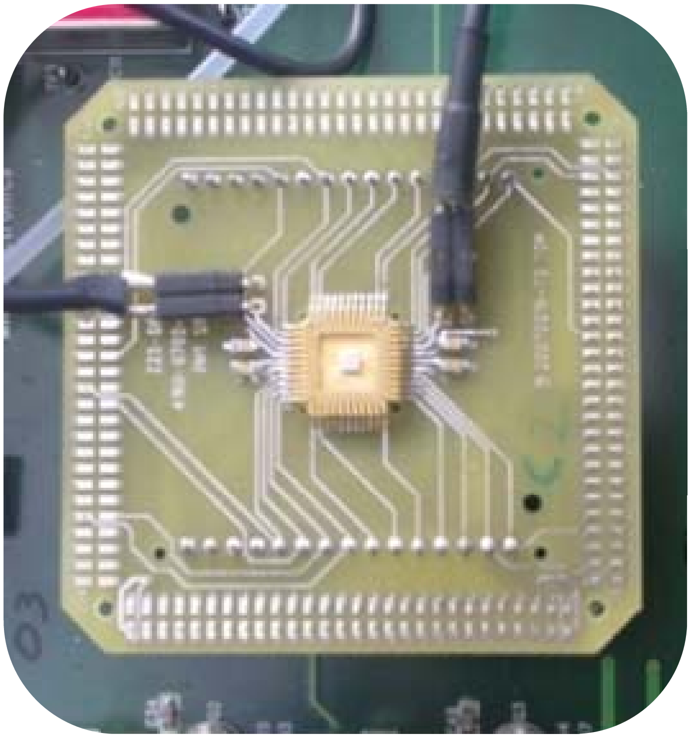

Figure 3.3: *Picture of the case study System on Chip soldered on a daughter board. The ceramic package was removed so to completely expose the silicon die to radiation.*

3.3 The Low-Cost Test Setup

A test interface board developed for manufacturing test was reused for radiation experiments sake. On-board voltage regulators supply the chip core and pads while a frequency modulator changes the operating frequency. The voltages may be changed with 0.02 V steps (1.2 V is the nominal chip supply voltage) and frequency from 15 MHz to 200 MHz with 5 MHz steps, being 20 MHz the nominal System on Chip working frequency. The motherboard (Fig. 3.4) provides complete connections to the chip functional/scan pins, therefore it affords full applicability of the silicon and radiation test through the diagnosis flows described in the previous Chapter. The motherboard structure is monolithic, which makes it very easy to handle and to place in the various radiation test facilities test chambers.

The high frequency clock source is placed on the bottom face of the motherboard, and the signal is directly sent to the System on Chip through BNC cables. On the contrary, the low frequency clock used by the DfT testing structure is sent by the host PC or controlling hardware. This solution permits to avoid any kind of long high speed connection. In fact, the high frequency source is close to the DUT and, thanks to the JTAG interface and the proposed built-in structures, a low speed connection with the

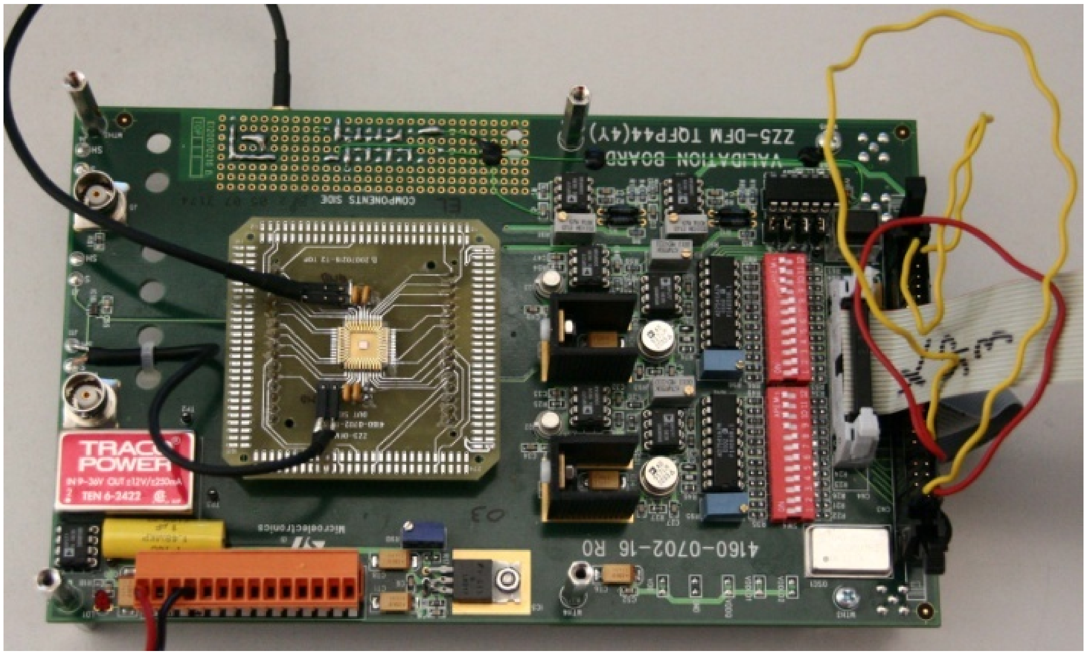

Figure 3.4: *The motherboard. Control circuitry change the voltage supplies of the DUT as well as the clock working frequency*

PC is sufficient.

The resulting test setup is depicted in Fig. 3.5. To perform radiation experiments on the developed SoC using the designed board just a host PC, a 5 signals parallel cable, and a 12 V external voltage source necessary to feed the overall system and regulators are needed.

The proposed test approach avoid the need of any high speed connection or expensive ATE to monitor the DUT. On the contrary, the setup is low-cost and very easy to handle. Previous works [Fra07] show how to have an easy to handle and monolithic testing board is very useful in radiation tests. A C++ procedure running on an host PC sends the JTAG signals to the DUT through a low speed parallel cable, to initialize the built-in structures. Once the built-in structures are activated and test is triggered, the DUT is stimulated by the on-board programmable high frequency clock source without the need of connections with the PC. When test is finished, results are downloaded from the DUT to the host PC through the tdo JTAG signal.

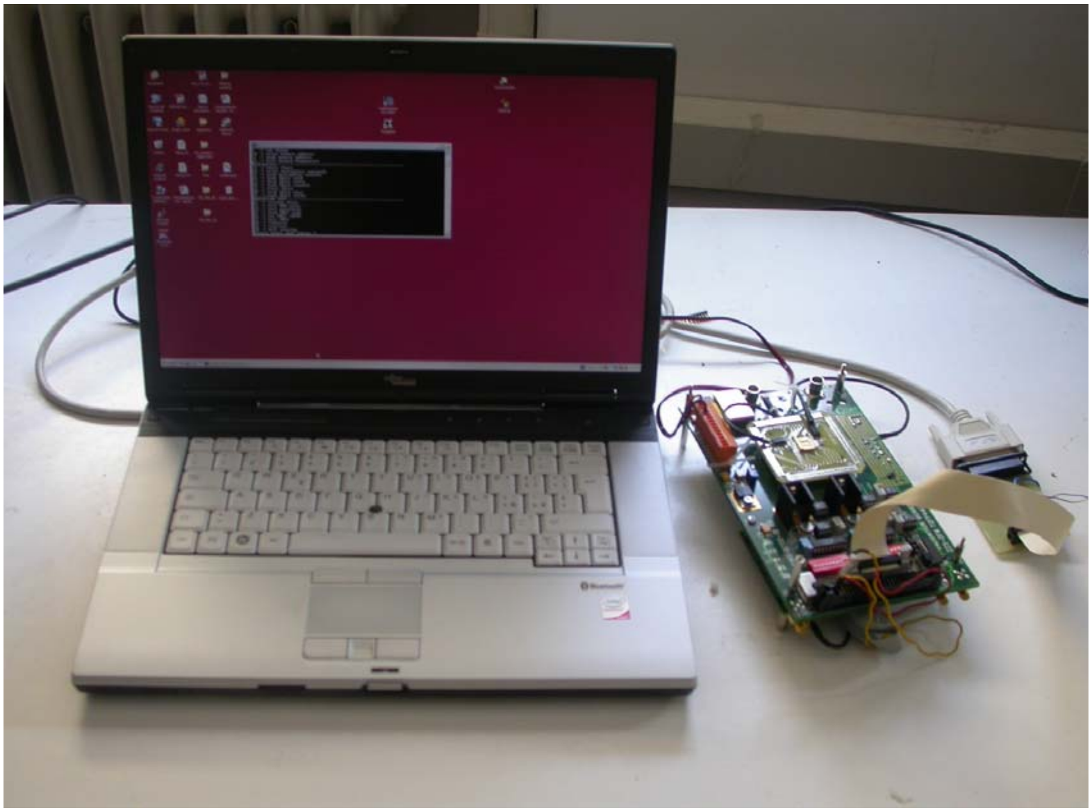

Figure 3.5: *The low-cost test setup used during radiation experiments .As can be seen, what is needed is just a host PC and a parallel cable. There is no high speed connection or expensive ATE needed*

Radiation testing grows in importance with the evolution of technology, especially in safety-critical application areas. Experiments performed on complex ICs such as SoCs are needed to analyze the radiation effects on real-world devices. We proposed a low-cost radiation test approaches based on the reuse of on-chip DfT logic, which provide precise information about failures.

The following chapters demonstrate the effectiveness of our approach and gives detailed radiation experiments results on the presented System on Chip.

Chapter 4

Embedded SRAM Radiation Test

Embedded memory cores often determine the yield in production processes of SoCs, as they are among the ones with highest integration density and tend to consume most of the transistors in SoCs [Coc94].

SRAM is the most widely used kind of memory because of its flexibility, high integration capability, and high density. Unfortunately, SRAM is very susceptible to radiation. As well documented in literature, an impinging particle may have enough energy to reverse the stored bit value, generating a bit-flip [Zie04] (known also as Single Event Upset). Several error detection and correction codes were developed to mitigate radiation and other undesirable source of errors effects. Unfortunately, with technology shrinking it may be possible for one single particle to interact with more than one SRAM cell, thus corrupting more than one bit, generating a Multiple Bit Upset, making ineffective most of the protection codes.

The effects of the radiation induced corruption of one or more bits in an SRAM embedded core in the overall System on Chip are various. The goals of radiation experiments on embedded SRAM are then mainly two. The first one is the static measure of memory arrays susceptibility to radiations in static conditions. The entire memory *array is exposed to radiation without being stimulated to let radiation induced errors to accumulate. After a predefined period of time, the array is scanned to detect bit-flips. This static test results in the measurement of the SRAM cross section (i.e. the probability of having a bit corrupted by radiation). To detect Multiple Bit Upsets, a dynamic test is necessary. The SRAM is initialized and then continuously checked for mismatches during radiation exposure. If the number of errors between two subsequent readbacks is increased of more than one unit, a Multiple Bit Upset may occurred.*

After a brief introduction on Radiation Effect in SRAM and how memory are typically tested and characterized, this Chapter described the proposed radiation test flow that takes advantage of the testing structure described in the previous chapters. Then, experimental setup is presented as well as the radiation test results.

4.1 SRAM Radiation Induced Effects

Since SRAM constitutes a large part of all advanced integrated circuits today, radiation effects and trends are of great important for chip manufacturers as well as critical applications designers and researchers. In fact, at the core of almost each modern digital system is a microprocessor that used a large embedded memory, usually SRAM.

SRAM cells are very susceptible to radiation, as an impinging particle may have enough energy to reverse the value of the stored bit. The device is not damaged by radiation, but the stored information is corrupted. Radiation is an issues for all space electronic devices, that are continuously hit by heavy ions, protons and other particles coming from cosmic rays or solar wind. Unfortunately, even at ground level radiation may disturb electronic. Alpha particles produced by chip, solder or package material and neutrons generated by the cosmic rays interaction with the terrestrial atmosphere may indeed affect the correct functionality of the devices. The radiation sensitivity of the SRAM strongly depends on the transistor characteristics, as the critical charge, defined as the amount of charge a particle must generate to produce a bit-flip, depends mainly on the cell critical nodes capacitance. Prior to the studies on the effects of SRAM corruption in the overall System on Chip functionality, it is fundamental to experimentally measure the memory core cross section, which is the probability of having a bit corrupted by radiation. Moreover, it is important to understand if the SRAM core is affected by radiation induced Multiple Bit Upsets (MBUs) that occur when one particle corrupt more than one transistor. In fact, the Error Correction Codes (ECCs) are typically effective in corrects just one corrupted bit per word. If two or more bits of one single word are corrupted simultaneously by the same impinging particle, most of the ECCs became useless.

Soft errors studies in memory cores importance grow as the technology is scaled and memory are enlarged. In fact, early SRAM core was more robust against radiation induced errors because of high operating voltages and higher node capacitance, which increase the critical charge to be depleted by the impinging particle to generate a bit-flip [Bau05]. With technology scaling, the SRAM junction area has been deliberately minimized to reduce capacitance, leakage, and cell area, while the SRAM operating voltage has been aggressively scaled down to minimize power consumption. However, with the latest modern devices, as feature sizes have been reduced into the submicron regime (lower than 0.25 µm), the SRAM error rate has saturated and may even be decreasing. This effect is primarily due to the saturation in voltage scaling, reduction in junction collection efficiency, and increased charge sharing. So, it is easier for a particle to reverse the stored value as there is less charge held in the node, but the active area is smaller, so it is less probable for that particle to hit it. This assumption also imply that an increase of Multiple Bit Upset is expected with technology scaling, as transistors sizes are reduced, but the particles do not, so the probability of having one particle to corrupt more transistors grows with transistor size reduction and density addition. In fact, scaling also implies increased memory density, so the overall SRAM based system number of errors increases. The exponential growth in the amount of SRAM in microprocessors has so led to the increase of radiation induced error rate with each generation with no end in sight.

As the System on Chip developed in collaboration with STMicroelectronics was designed to be part of an automotive project, we decide to focus our attention on terrestrial environment, taking both neutrons and alpha particles effects into account. The following paragraphs describe the efficient testing strategy we develop to measure the SRAM core radiation sensitivity as well as the radiation experiments results we obtained using an Americium alpha source available at LNL and the pulsed neutron beam available at the ISIS facility.

4.2 Proposed Test Flow

In the manufacturing test flow, memory test and diagnosis usually rely on the execution of a series of diagnostic March algorithms on the memory array, and allow to spot failing locations and determine the type of the discovered faults. For the latter concern, a very useful feature is the programmability of the memory test algorithm to be applied, which is a prerogative of programmable BIST (pBIST) architectures.

The flexibility of the programmable diagnostic BIST approach can be fruitfully exploited for radiation testing, as described in Chapter 2 of this manuscript. The programmable BIST engine can be used to write specific configurations into a memory to prepare the array for the radiation experiment, and then for reading out bit-flip positions, without requiring direct write and read operations from the external tester. Memory content setup and result retrieval is very fast even compared to the soft-BIST methodology. The employed flow consists of three steps to be iterated when the SoC is exposed to radiation:

1. memory content preparation using the programmable BIST running a simple pseudo-March algorithm, e.g., {↑(w0)}
2. wait for a defined period
3. readback via programmable BIST

The SRAM is initialized by writing a known pattern on the entire array. This is performed sending a high-level instruction to the pBIST through the JTAG interface. For radiation tests sake just simple march algorithm may be used. As explained in Chapter 2, complex March algorithms are used to detect functional faults in the manufacturing tests processes. Examples of possible and common Functional Faults are: cell stuck, driver stuck, Read/Write line stuck, shorts between data lines or crosstalk in data lines, and others. Functional Faults detection is a manufacturing test challenge, and research studies are being performed to find out the most efficient way to detect those faults using specific March algorithms. For radiation tests sake, the memory array to be tested under radiation must be perfectly working. We so perform various March tests on the devices prior to radiation exposure so to be sure that the array is not defective. Regarding March test to execute during radiation tests, we know that radiation induced errors on SRAM cells are just bit-flips, and thus very simple March algorithms are sufficient to detect them.

We performed our experiments using tree different test patterns: *All 0s* (00), *All 1s* (FF), and *Checkerboard* (AA). In the first case the memory content preparation may be done using the simple March algorithm {↑(w0)} that reset all the available bits in the entire array; in the case of *All 1s*, the {↑(w1)} is used and in the latter case a combination of the two March algorithms. After the execution of the March algorithm the SRAM core is exposed to radiation for a predefined time without being stimulated so to let errors to accumulate. Radiation exposure time must be long enough to collect a statistically significant number of errors but sufficiently short so to be reasonably sure that any memory cell is corrupted at most once. A best compromise for exposure time can be estimated knowing the expected error rate from similar tests on same technology devices and eventually adjusting it on the field with a calibration test run. When the exposure time elapses, the entire array is scanned to detect mismatches. This operation is again performed by the pBIST that is programmed trough the JTAG interface with {↑(r0)} March algorithm for *All 0s* test, {↑(r1)} for *All 1s* test, and a combination of them for the *Checkerboard* test.

The only information exchange between the host PC and the BIST regards the pattern to be written and errors detected, permitting the use of simple and low-cost connections. The DUT may be exposed to radiation even during memory content preparation and readback procedures as these processes are handled by hardwired circuitry at high frequency and, as reported in the following paragraphs, it is very unlikely for alphas to corrupt a bit during their executions.

4.3 Experimental Setup

In order to estimate the radiation sensitivity of the embedded SRAM core and validate the proposed strategy, we performed a set of radiation testing experiments with an Americium alpha source at Laboratori Nazionali di Legnaro, INFN, Italy and with an accelerated pulsed neutron beam at ISIS, Rutherford Appleton Laboratory, Didcot, UK.

4.3.1 Radiation sources

The System on Chip we tested was developed by STMicroelectronics as a test chip to study an eventual application in automotive projects. As stated above, ground level electronics functionality is mainly affected by alpha particles and neutrons. We performed radiation tests both using an Americium alpha source and a pulsed neutron beam.

Alpha source

The first accelerated radiation tests campaigns were performed with an Americium source emitting alpha particles. The shape of the ^{241}Am deposit is circular, with 4 mm radius and the source activity is 3.3 kBq. The Americium is deposited on a stainless steel disk and encapsulated in a perforated plastic package which can be easily handled and placed on the chip under test. The half-time of ^{241}Am is 433 years, so the source can be modeled as a constant flux emitter. Alpha emission from the source is isotropic, therefore particles reach the die with different angles.

Pulsed neutron beam

A second radiation test experiment was performed at the VESUVIO beam line at ISIS. VESUVIO is commonly employed for condensed matter studies, exploiting neutrons above 1 eV, the so-called epithermal neutrons. VESUVIO monitor detectors provide information about the low-energy neutron fluence hitting the irradiated samples (ISIS neutron flux is of $7.86 \cdot 10^4 n/cm^2/s$) [Vio07].

4.3.2 Radiation test protocol

We used the same test protocol for radiation experiments with the Americium source and with the ISIS pulsed neutron beam. The employed March-based programmable BIST described in Chapter 2 is able to execute any test program for SRAM cores. This BIST architecture includes the following features that were fundamental for the radiation experiment sakes:

- March-based BIST programmability. The March algorithm microcode is stored in a dedicated memory (256 bit)
- internal and autonomous microcode correctness check
- diagnostic registers for downloading failing location information [App03]
- synchronous and asynchronous BIST reset schema

The employed radiation experiment flow and the time employed for the execution of each step are reported, considering a core/BIST frequency of 20 MHz and serial data transfers at 30 KHz. Higher test frequencies are not needed in this setup since SEUs effect are concerned.

1. **memory content preparation**
   a. load short pseudo-March code into pBIST Memory (64x4 bits), e.g., ↑(w0) **22 ms**

b. test March code through MISR **0.7 ms**

c. pBIST reset and parameters set-up **3.3 ms**

d. BIST execution (memory writing) **3.9 ms**

2. **wait for a specified time** (depending on the radiation source)

3. **readback via programmable BIST**

a. load short pseudo-March code into pBIST Memory (64x4 bits), e.g., ↑(r0) **22 ms**

b. test March code through MISR **0.7 ms**

c. pBIST reset and parameters set-up **3.3 ms**

d. BIST execution (memory reading) **3.9 ms**

e. extract information about an error; if needed
repeat from (4.b) **2.8 ms**

The process is iterated many times to collect data varying operating parameters such as supply voltage to study device sensitivity dependences. We tested the embedded SRAM with different supply voltages, varying it from 1 V to 1.3 V (being 1.2 V the DUT nominal voltage), and with different test patterns (*All 0s*, *All 1s*, and *Checkerboard*).

The errors accumulation time strongly depends on the radiation source used and DUT sensitivity. It is fundamental to tune the accumulation time so to gain from the test a statistically significant number of errors. However, the accumulation time should not be too long to avoid the possibility of having an already corrupted bit reversed by a second impinging particle. Having 50 errors per run seems a good tradeoff between the conflicting needs. An expected error rate can be calculated studying similar technology and structure SRAM radiation tests presented in literature. Thanks to the flexibility of our testing protocol, the accumulation time can also be refined in the field, with a set of calibration run.

4.4 Experimental Results

We have performed alpha radiation experiments on July 2008 and with neutrons on March 2009. The neutron induced error rate is definitely lower than the alpha induced one, as expected. Here we report a detail description and analysis of the alpha experiments result and a comparison with neutrons experiments. In both cases, our strategy was fruitfully applied, and tests were very easy to prepare.

4.4.1 Alpha test results

We perform alpha radiation experiments in air, simply placing the Americium source very close to the SoC silicon die and the host PC in the proximity of the motherboard. As described in Chapter 3 (Fig. 3.3) devices destined to alpha experiments were built in a ceramic package which can be easily removed, thus completely exposing the silicon die. Moreover, as the SoCs are soldered in a daughter board, without the need of any socket, the radiation source can be placed just few millimeters away from the device active area. As tests are performed in air it is fundamental to minimize the

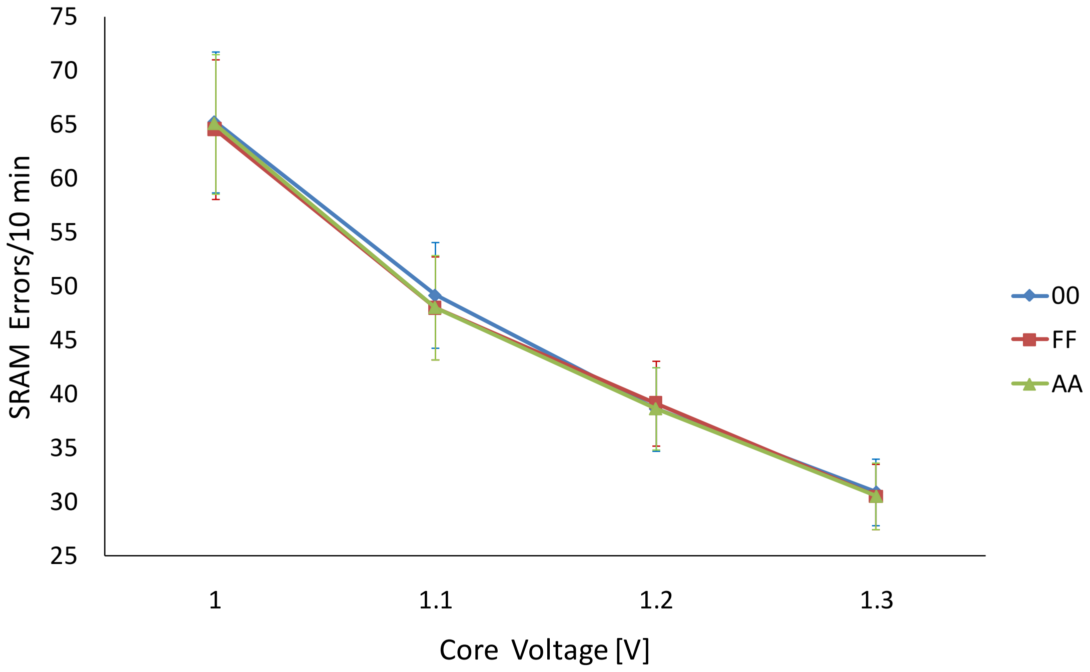

Figure 4.1*: Radiation induced errors in the embedded SRAM array per run as a function of core voltage supply and test pattern.*

Table 4.1: *Experimentally observed errors per run affecting the embedded SRAM array as a function of test pattern and core voltage.*

Pattern	1.0 V	1.1 V	1.2 V	1.3 V
All 0s	65.3	49.1	38.6	30.8
All 1s	64.6	48.0	39.1	30.5
Checkerboard	65.1	48.0	38.6	30.5

distance between the DUT and Americium source, as demonstrated in [Bau07].

Experimental results gathered using the Americium alpha source are depicted in Fig. 4.1 and reported in Tab. 4.1. We have tested the embedded SRAM core with different patterns and varying the supply voltage. As described in paragraph 4.2 we performed experiments initializing the array with *All 0s*, *All 1s*, and C*heckerboard* patterns. Thanks to the available motherboard circuitry, we were allow to change the DUT supply voltage with 0.02 V steps. We decide to perform tests at 1 V, 1.1 V, 1.2 V, and 1.3 V, being 1.2 V the nominal voltage supply of the SRAM core. For each of the patterns and voltages we performed hundreds of runs under radiation. As reported in Tab. 4.1 and shown in Fig. 4.1, the number of errors per run affecting the embedded SRAM grows as core supply voltage decreases. This is in agreement with previous tests performed on SRAM and reported in literature [Hei07]. On the contrary, no particular differences were found in SRAM sensitivity between different test patterns for a given voltage. This is due to the structure of the Bit Cell composing the embedded SRAM array, which is perfectly symmetric and balanced.

4.4.2 Neutron test results

The beam availability at ISIS facility was of 4 days, sufficient to gather just few hundreds of bit-flips in the entire memory array. Because of the limited beam-time availability and the low error rate of our device, it was not possible to perform all the experiments carried out in the case of alpha particles. Our intention was to prove the effectiveness of our testing structure even when neutron beam is concern. The main issue in the case of neutron test is that the controlling hardware (i.e., the host PC in our case) must be placed outside the irradiation room to prevent neutrons to corrupt it (see paragraph 2.2, Fig. 2.1). The distance between the host PC and the DUT is almost 5 meters, so a high frequency connection between them will be very expensive and hard to maintain.

A second issue we faced regards accumulation time. To obtain a significant number of errors (almost 5), 2 hours of neutrons exposure was necessary. The test setup was working for 4 days, without losing functionality, attesting the robustness of our strategy. The main advantage, in this case, is that during the 15 minutes of errors accumulation, the structures and host PC are not working. In fact, the pBIST can be programmed with the error detection March code when the accumulation time elapsed, and no other operations are needed.

The measured error rate, with an average proton current of 150 μA [Vio07], is of 2.92 errors per hour. We have also calculate the Failure In Time (FIT) of the device, which is the number of failures (bit-flips in the case of radiation issues) that can be expected in one billion device-hours operation. To calculate the FIT we started from the experimentally obtained cross section, which is the number or radiation induced errors normalized to the number of exposed bits, divided by the particles fluence (number of particles that hit the device per time unit). FIT are the calculated multiplying the DUT cross section with the natural particles flux. Considering alphas, 0,002 $\alpha/cm^2/h$ is the emission expected by Ultra Low Alpha (ULA) packaging components, while the neutron natural flux vary with altitude and is of about 20 $n/cm^2/h$ at sea level. Precise data cannot be given in this manuscript because of NDA agreement with the DUT manufacturer, but we can attest that the memory core evaluated FIT is about 7 times higher with respect to neutron FIT, with the given alpha emission and neutron flux.

4.4.3 Multiple Bit Upsets

Besides the number of errors affecting the memory array, the pBIST gives information on errors location and corrupted data. This permits to draw the map of errors shown in Fig. 4.2, where each white point indicates a bit that has been corrupted at least once by alphas during the overall experiments campaign. From the figure it is clear that the SRAM array is uniformly affected by errors. Information retrieval relies on the BIST structure, that after being programmed checks the memory array and sends the number of errors, their values and locations to the host PC via parallel port using the JTAG interface. A word of 22 bits must be read for each error to have complete information on it. Parallel port signals cannot be faster that some KHz and bits are sent serially from the device. However, this protocol minimizes the communication between host PC and BIST and the time needed to retrieve information is about 6 ms for each error read. In the worst case Americium radiation corrupts a bit in the DUT every 8 seconds, so we can state that no errors occur in the memory array during initialization and readback.

Information about error location is very important to detect Multiple Bit Upsets. Our testing strategy can be then fruitfully applied also to dynamic SRAM radiation test, performed reading continuously the array. If between two consecutive checking the number of detect bit flips is incremented by more than one, an MBU may be occurred. With the information about error location given by pBIST and memory physical organization by the core manufacturer, we may understand if the corrupted bits are logically close to each other and belong to the same word.

4.4.4 pBIST criticality

The main disadvantage of the proposed testing strategy is that being built-in, all the DfT structures are exposed to the same radiation flux as the DUT, and then may be corrupted. However, as reported in paragraph 3.1, DfT architecture area is just the 2,57% of the overall system area. The probability for a particle to hit a DfT structure is then definitely lower than the probability to hit the tested cores.

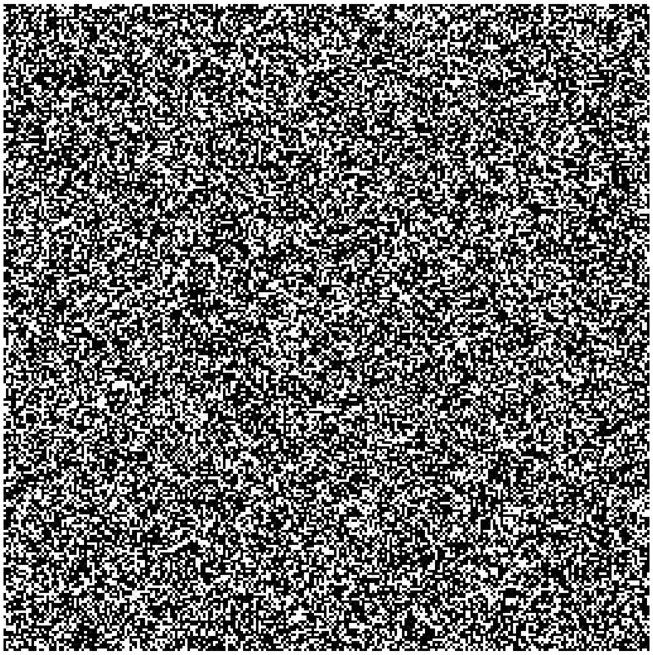

Figure 4.2: *Radiation induced errors map. Each white point represents a bit that has flipped at least once during the test campaign.*

In the particular case of SRAM radiation test, the probability of having the testing structure corrupted is very low. In fact, the pseudo-March tests used in our experiments are very simple, and consist of less than 10 instruction of 4 bits each. Even if these instructions are stored in SRAM cells, it is very unlikely for them to be corrupted by alphas. As the code memory is built using similar bit cells as to the ones composing the tested embedded SRAM, its radiation induced error rate is very similar to the memory core one. Moreover, we know that only 40 code memory bits are used to store the very simple March algorithm used in our radiation tests. Their corruption affects the tests only if it happens during the pBIST programming phase (1.a or 3.a in the test schema reported in paragraph 4.3.2), which last 22 ms, March code testing through MIST (1.b or 3.b), 0.7 ms, pBIST reset and parameters set-up (1.c or 3.c), 3.3 ms, BIST execution (1.d or 3.d), 3.9 ms, and, eventually, errors information extraction, which may vary from test to test (168 ms in the worst case of 60 errors per run). The overall pBIST criticality then last less than 30 ms for memory content preparation and at most 2 μs for errors detection and information retrieval. With the above explained assumption and timings we can estimate that the probability for errors to occur in the code memory is 4 orders of magnitude lower than in the embedded SRAM array. Moreover, on-line code memory test is periodically performed to avoid mistaken results. We have found no discrepancy in the code memory in any of the performs experimental run.

4.5 Conclusions

Radiation testing on SRAM grows in importance with the evolution of technology, especially in safety-critical application areas. Experiments performed on SRAM core embedded in complex ICs such as SoCs are needed to analyze the radiation effects on real-world devices. We proposed and demonstrated the effectiveness of low-cost radiation test approaches based on the reuse of on-chip DfT logic, which provide precise information about failures, applied to a case study embedded SRAM core.

Chapter 5

Embedded Microprocessor Radiation Test

This Chapter presents and discusses the results of alpha Single Event Upset (SEU) tests on an embedded 8051 microprocessor core. Different resources available in an embedded microprocessor may be corrupted by radiation, and the effects at the output an on the overall system functionality may be various. Our intention is to measure the radiation sensitivity of the different internal microprocessor-based SoC resources and try to understand how their corruption affects the output.

Fault injection is a powerful and helpful tool in understanding errors propagation, for instance, and their effects at the output, but has some shortcomings [Car02]. Fault injection does not provide a direct extrapolation to operating conditions and, generally, does not account for possible variations in the radiation sensitivity of the different memory bits inside a complex design. The experimental assessment of complex chips with radiation sources is therefore of primary importance. In this Chapter, we present an experimental analysis of the sensitivity of a modern embedded processor, combining alpha irradiation and analytical calculation of derating factors.

In particular, Code RAM and User Memory were tested employing a simple March-test. Each memory location is initialized writing a known pattern, then the *program waits for a specified time letting errors to accumulate, and checks for mismatches. Internal user registers sensitivity was measured loading a code in the microprocessor that resets each register and then enters an infinite loop. After a specified time, an interrupt subroutine is activated, and checks each register for errors. The number of corrupted bits is then sent to the output ports that are monitored by MISR circuitry. Finally, we used this data to estimate benchmark codes alpha sensitivity basing on memory resources needed for their computation, and these estimations were compared to experimental results.*

5.1 Microprocessors Radiation Induced Effects

Microprocessors are very complex devices, composed of both logic and memory resources. Their test is a major challenge for the research community and for manufacturers. As long as radiation is concern, memory resources may be corrupted, as described in the previous Chapter of this manuscript. The effect of the bit-flip in the microprocessor execution depends on various factors. If the corrupted resource is not used, or if the data it stores is obsolete, for instance, there will be no effect at the output. On the contrary, the corruption of the Program Counter register may result in a compromised program flow, or halting condition. Moreover, also the code RAM may be corrupted, and if this happens, a wrong instruction may be fetched, generating unpredictable results.

Unfortunately radiation, besides corrupting registers and memory resources, may also disturb logic. In fact, when a particle struck a node, it may generates a Single Event Transient (SET), which is a temporary voltage pulse that may propagate, and be latched in a memory element, leading to Soft Error [Bau05]. Working frequency has a major impact on SETs capture: the higher the frequency the larger the probability of having a memory element corrupted by a propagating transient [Dod04][Eat04]. It is then fundamental for tests to be performed at the operating frequency, to avoid SET underestimation.

Thoroughly testing a microprocessor under radiation is then an expensive and time-consuming task. When performing radiation tests on complex devices, as a microprocessor, many different resources may be affected by errors, but these do not necessarily appear at the output. Checking a microprocessor user memory after a test program execution, for instance, may not be sufficient to characterize its sensitivity. It would be very attractive to understand the sensitivity of each resource, which errors affected the device computations and which ones were masked, to extend the results collected during the radiation tests to other conditions. These data permit to predict a device sensitivity as well as a program failure rate. On the other hand, knowing which resources are more likely to fail and how errors propagate gives indication on hardware/software designing rules for lowering device and running program sensitivity.

Radiation tests are performed using radioactive sources or facilities that accelerate heavy ions or produce neutron beams. Different constraints may be imposed to the test set-up. For instance, the DUT may have to be placed in a vacuum irradiation chamber and high-speed connections may have to be run for several meters and across flanges, making the test preparation quite challenging and expensive (see paragraph 2.2). It is then very attractive to limit the number of cable connections and the speed of the information exchange between the DUT monitoring circuitry and test equipment (host-PC, for instance).

5.2 Proposed Test Flow

As described in Chapter 2, processors manufacturing test and diagnosis may be based on ATPG-generated patterns applied through scan chains, or on the execution of Software-Based Self-Test (SBST) procedures [Ber05][Kra05]. The latter methodology consists in making the processor run a suitably developed program able to excite faults and propagate their effects to observable points. The test program runs at-speed in normal operational conditions, hence no architectural modifications of the processor are needed. The SBST application procedure encompasses the following operations:

- uploading of the test program in a suitable memory area
- activating its execution, i.e., letting the program run, stimulating the components
- retrieving the results opportunely stored during the test program execution

Dedicated Infrastructure-IPs (I-IPs) may be usefully integrated in the SoC [Ber05], which support the application of SBST methodologies and provide efficient interfaces to the outside. Their main tasks are the management of code upload by interacting with the system bus, the launch of the test program execution (e.g., activating either reset or interrupt signals) and the compression of the test result through Multiple-Input Signature Registers (MISR). For radiation testing, SBST supported by I-IP allows effectively and easily interacting with the processor, minimizing the amount of exchanged data and therefore allowing low-bandwidth communication with a less expensive external test management unit.

Finally, standard communication infrastructures and protocols such as the ones defined by the IEEE 1500 [IEE05] and IEEE 1149.1 [IEE94] standards provide core isolation, separation of test execution and data transfer frequency domains and serial test data transfer, facilitating the experimental set-up of radiation experiments.

5.3 Experimental Setup

We performed alpha radiation tests on the test vehicle manufactured by STMicroelectronics in a 90 nm technology, embedding the 8051 microprocessor described in Chapter 3. The embedded SRAM core included in the SoC is used both as code RAM and user RAM by the microprocessor. It is possible to achieve high diagnosability for each of the cited components resorting to an Infrastructure-IP (I-IP), which manages the execution of at-speed SBST procedures on the processor, a March-based programmable diagnostic BIST (pBIST) exploited for memory test, and additional scan structures inserted for the sake of observability and controllability of the final test.

IEEE 1500 wrappers are inserted in order to provide a low-frequency common interface to the test logic, and permitting to the external test controller to interact with the on-chip devices by sending high-level instructions. The test structures are then accessed through IEEE 1149.1 (JTAG) TAP controller, thus relying on only 5 signals. A C++ software tool running on a host PC gives the JTAG commands through a parallel port, collects and stores the test results, and controls the voltage regulators and the frequency modulator.

5.3.1 Radiation source

The test chip was developed by STMicroelectronics to be part of a terrestrial application, and thus just alpha and neutron radiation effects are to be concerned. In Chapter 4 we demonstrated how the error rate due to the pulsed neutron beam available at ISIS, RAL, Didcot, UK is definitely lower than the Americium induced one. As the available memory bits number is definitely lower in the microprocessor core than in the

SRAM core, the number of bit-flips in the microprocessor's memory resources is going to be definitely lower than the number of bit-flips in the overall SRAM core. To have a statistically significant number of errors, then, a long exposure time may then be necessary. Unfortunately, the beam availability at ISIS facility is too restricted to observe a satisfactory number of events. The best solution is to use a more active alpha radiation source.

Single Event Transient may also affect the microprocessor functionality but, as we will see, their contribution in the overall error rate is negligible as long as 90 nm CMOS technology irradiated with neutrons or alphas is concerned [Shi02].

In order to determine the DUT sensitivity to alpha particles, we then performed a set of radiation testing experiments with an Americium alpha source at DEI, Università di Padova, Italy. The shape of the ^{241}Am deposit is square, 35 mm wide, and the source activity is 250 kBq. The Americium is deposited on 4 active strips covered with 2 µm of gold-palladium mounted on a stainless steel support which can be easily handled and placed on the chip under test. The half-time of ^{241}Am is 433 years, so the source can be modeled as a constant flux emitter. Alpha emission from the source is isotropic, therefore particles reach the die with different angles.

5.4 Static Test

The first step in the characterization of a complex device as a microprocessor, is the static test, to measure the radiation sensitivity of the available memory resources. As described in Chapter 3, the microprocessor memory resources are composed of internal registers, code RAM and user RAM. The static test aims at calculating the static cross section of these resources.

5.4.1 Static test protocol

The microprocessor uses the embedded SRAM core both as code RAM and user RAM. The core has been tested using the protocol described in detail in paragraph 4.3.2. Briefly, the pBIST initialized the entire memory array with a known pattern, then for a specific period of time the device is exposed to radiation so to let errors to accumulate. When that time elapses, the pBIST is programmed to check for mismatches and results are sent to the host PC through the JTAG.

To measure the sensitivity of the internal registers we took advantage of the buil-in Software Base Self Test. We upload an ad-hoc application designed to reset (or, eventually, set) each accessible register in the microprocessors and then enters an infinite loop. After a predefined time during which the device is exposed to radiation without being stimulated to let errors to accumulate, an interrupt subroutine is externally activated and checks for corrupted bits, sending the number of detected mismatches to output ports that are downloaded through the JTAG.

5.4.2 Static test results

The device we tested includes an SRAM core that consists of 512Kbit of symmetric cells which is used by the microprocessor both as code memory and user memory. We used a simple march algorithm to write a known pattern in the entire array, wait for errors to accumulate, and check for mismatches as described in Chapter 2. As the Americium source we used for the microprocessor test is definitely more active with respect to the one used for the embedded SRAM core described in the previous Chapter (250 kBq in the former case, 3.3 kBq in the latter), the accumulation time in the former case is going to be shorter. The best tradeoff between high number of errors collected and low probability of having a corrupted location flipped by a second impinging particle was found to be 6 minutes, so to gain about 50 errors per run.

The alpha sensitivity of the microprocessor code RAM and user RAM, i.e., the number of errors observed normalized to the available bits and unit time, is reported in Tab. 5.1. We measured an average error rate of $2.07 \cdot 10^{-6}$ errors per bit per time unit under alpha irradiation.

Regarding the internal registers, we know that the microprocessor utilizes 1669 flip-flops for computation, 1208 of which are directly accessible. We measured sensitivity to be $1.51 \cdot 10^{-6}$ errors per bit per time unit for the accessible registers (Tab. 5.1). As the remaining internal registers are built in the same technology and with the same libraries, we can assume their sensitivity to be similar to the accessible ones.

Table 5.1: *code RAM, user memory, and registers experimentally calculated static cross section.*

Resource	Errors per bit per time unit
Code and User Memory	$2.07 \cdot 10^{-6}$
Registers	$1.51 \cdot 10^{-6}$

The difference between code SRAM and registers error rates is attributable to the different structure of the cells composing the different resources. The SRAM core is dense and thick as spatial occupation is a major concern, while registers are distributed and spread in the microprocessor. For this and other reasons, register cells are usually built and designed differently from SRAM ones, that is why their radiation sensitivity may differ. Various previous works showed and described in detail how radiation sensitivity varies with different designing rules and building libraries [Hei07]. The predicted error rate of each loaded code should be calculated taking into account all the different contributions, i.e., code ram errors, register errors, logic errors. Registers are less sensitive to alphas with respect to SRAM, and moreover, their contribution to the DUT overall error rate should be normalized taking into account the effective number of registers involved in the code execution with respect to the number of code RAM and user memory ones, that is likely higher.

This result is very important and should be used when building a fault injection platform [Per08]. The probability function used to inject an error in a specific memory location should take also into account the different sensitivities the different resources may have. In this particular care, the probability of having a code RAM or user RAM bit corrupted should be slightly higher than the one of having a register bit flipped.

5.5 Dynamic Test

To understand the microprocessor behavior when exposed to radiation, the static test is not sufficient. The main issue is that having a bit corrupted is not a necessary nor a sufficient condition of having an output error. In fact, on one side the corrupted bit may not be used, the data it stored may be obsolete, or its corruption may have no effect on the output. On the other side a Single Event Transient may be produced by radiation and compromise the microprocessor execution. It is then fundamental to test the microprocessor under operating condition, exposing it to radiation while executing an application.

As it is very unlikely to have SET produced by alpha in a 90 nm technology, we focused our attention on the memory resources corruption contribution to the system overall error rate. We designed two benchmark codes, one maximizing internal registers usage and the other maximizing the code RAM usage to see and understand how radiation induced errors affecting the different resources are propagated or masked.

5.5.1 Dynamic test protocol

The buil-in structure described in details in Chapter 2 can be fruitfully applied to the dynamic test of the microprocessor. Through the JTAG, the code to be executed is uploaded at low speed in the code RAM memory of the microprocessors. When test execution is triggered by the host PC running code, the SBST executes the uploaded code at working frequency, without the need of any external assistance. As the high frequency clock source is placed on-board, there is no need of having high speed expensive connections. While the device is exposed to alphas and the test code is being executed, the MISR circuitry continuously monitors the output ports to detect any output error. When test is finished, MISR signature can be downloaded at low speed through the JTAG and the host PC checks for its correctness. The 32-bit wide MISR module ensure a low percentage of aliasing, and is then very effective at monitoring the codes execution. A watchdog is also available so to monitor the program flow, and detect any timeout.

The working frequency of the DUT can be easily varied from 15 MHz to 200 MHz thanks to the on-board programmable clock source. The number of SETs generated by radiation strongly depends on the combinational circuit working frequency, while SEUs number remains constant. If the observed number of output errors remains constant at the different tested frequencies, we may conclude that SETs contribution is negligible with respect to SEUs one. We performed tests at different frequency, and see no significant variations in the radiation induced output error rate of our device. In the following paragraphs details about tests performed at 20 MHz, which is the nominal DUT working frequency, are reported.

5.5.2 Tested algorithms and codes

In order to estimate the sensitivity of the device under dynamic operating conditions, we implemented different codes as test benchmarks, including:

***ADD_Loop*:** a loop of 255 sums. The results are continuously sent to output ports and checked by MISR.

***255_ADD*:** a sequence of 255 sums (without loops). Again, the results are sent to output ports and checked by MISR.

The differences among these algorithms rely both in the number of instructions needed to implement the algorithms (and so code memory usage) and in the number of internal register bits needed to execute them. Our intention is to understand, thanks to our low-cost test setup, how internal registers and code RAM corruption affects the microprocessor executions. To do that we emphasized the code differences maximizing the number of registers usage and minimizing instructions in the ADD_Loop algorithm while minimizing registers and maximizing instruction in the 255_ADD algorithm.

The assembly code that implements the ADD_Loop algorithm is the following:

```
MOV A, #000h
MOV R1, #000h

LOOP:

        MOV A, R2
        ADD A, #001h
        MOV R2, A
        MOV A, R1
        SUBB A, #001h
        MOV R1, A
        JNZ LOOP
```

The accumulator stores the partial sums results, while register R1 is used to take trace of the loop steps, and register R2 is used as a temporary backup for the accumulator values. The only instruction profitable for the computation is the framed one, all the others are needed to implement the loop.

In the case of 255_ADD algorithm, the assembly code is very simple and consists of a sequence of 255 *ADD A, #0001h* instructions. In both algorithms each sum

Table 5.2: *Resources needed to implement ADD_Loop and 255_ADD algorithm*

	ADD_Loop	**255_ADD**
Instructions	9	255
Total Code SRAM bits	144	4128
Registers	3	1
Total flip-flops	207	191

corresponds to the assembly instruction *ADD A, #0001h* that increments the Accumulator value by 1. At the computation end the correct result should obviously be #0FFh (255 in decimal) and each mismatch indicates that radiation caused an error somewhere in the microprocessor. It's easy to see (Tab. 5.2) that more registers are involved in the computation of ADD_Loop code with respect to 255_ADD, as R1 and R2 register are needed to permit the loop execution. In both cases 175 internal flip-flops are used for the computation of the *ADD A, #0001h* instruction. The second important difference among the two codes is the memory necessary to store their implementations. In fact, while just 9 instructions are needed to implement the ADD_LOOP algorithm and 255 to implement the 255_ADD one.

To detect any kind of errors in the computation, the sum results are continuously sent to the output ports monitored by the MISR circuitry, thus in both cases two additional instructions are needed and output port registers are also used. Moreover, a hard wired watchdog ensures the detection of radiation-induced errors that lead to system halting or generate infinite loops.

5.5.3 Dynamic test results

The first step in the dynamic cross section measurements is to predict the number of alpha induced bit-flips affecting the different memory resources during the two benchmark codes executions. This can be simply done correlating the static test experimentally obtain cross sections with the code analysis of paragraph 5.4.2 summarized by Tab. 5.1 and graphically represented in Fig. 5.1. To compute an operation, as the ADD one, the microprocessor utilizes 175 internal flip-flops (e.g. the

Table 5.2: *ADD_Loop and 255_ADD used resources and errors per execution and accumulation time in code RAM and register bits*

	ADD_Loop	**255_ADD**
Code SRAM bits	144	4128
Errors per execution	$6.08 \cdot 10^{-7}$	$2.91 \cdot 10^{-6}$
Errors with accumulation	$1.69 \cdot 10^{-4}$	$48.08 \cdot 10^{-4}$
Register bits	207	191
Errors per execution	$6.39 \cdot 10^{-7}$	$9.88 \cdot 10^{-8}$

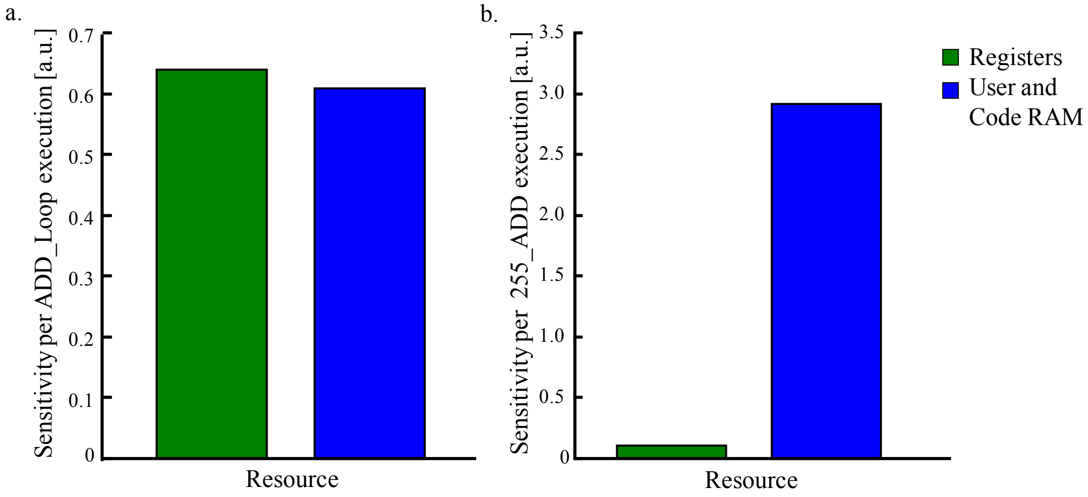

Figure 5.1: *ADD_Loop (Fig.5.1a on the left) and 255_ADD (Fig. 5.1b on the right) expected alpha induced bit-flips in the different memory resources used in the codes execution. In the case of ADD_Loop, the expected bit-flips number affecting used registers is slightly higher than the number of errors in the used code RAM. On the contrary, in the case of 255_Add, the number of expected errors in the used memory resources are dominated by code RAM bit-flips*

PC, timer controls, Interrupt control, etc.). In the execution of ADD_Loop code 4 additional registers of 8 bit each are involved, i.e., the Accumulator, R0 needed for loop step check, R1 as temporary Accumulator value storage, and P0 for outputs checking. The probability of having one bit corrupted during execution time (operating frequency is 20 MHz) among calculation involved registers is then $6.39 \cdot 10^{-7}$. On the other hand, only 144 bits are needed to store the code instructions, thus the probability of having a corruption in those bits during code loading and execution is $6.08 \cdot 10^{-6}$. In the execution of 255_ADD code less registers are needed for computation, as there is no loop implemented. The only registers used are the Accumulator and P0. The probability of having one register bit corrupted is $9.88 \cdot 10^{-8}$ during execution time. As there are 255 *ADD A, #001h* instructions loaded in code RAM, 4128 bits are needed to store the entire code instructions, so the probability of having a corruption in those bits is $2.91 \cdot 10^{-6}$, definitely higher than the registers one.

From Fig. 5.1 a. and b. it is clear that while in the case of ADD_Loop (Fig. 5.1a), the expected bit-flips number affecting used registers is slightly higher than the number of errors in the used code RAM, in the case of 255_Add (Fig. 5.1b), the number of expected errors in the used memory resources are dominated by code RAM bit-flips. Finally, looking at the different graph scales, the probability of having a memory bit corrupted during 255_ADD execution is reasonably higher than to have a bit corrupted during Loop_ADD execution.

However, these values, reported in Tab. 5. and Fig. 5.1, are just a rough overestimation of the effective code sensitivity. Code RAM bits, in fact, are not critical during the entire code execution. In the case of 255_ADD, once the instruction has been fetched, its corruption in the code memory array is completely irrelevant for the output correctness. On average, one instruction in this loop is critical for half the execution time of this application. ADD_LOOP code instructions, on the contrary, are repetitively processed, and thus are critical till the end of code execution. Moreover, the predicted error rates for the two test benchmark codes do not consider radiation induced error rates in the logic resources. The SET error rate in the case of a 90 nm technology node, however, is definitely lower that the SRAM one and thus may be neglected [Sce02].

In a typical application, the microprocessor is not only exposed to radiation while code is executed. This may have serious consequences, as errors accumulate during exposure time. Errors accumulation, however, affects only the code RAM, as all the registers are typically reset prior to code execution. To have a realistic error rate, we left errors to accumulate in the code RAM prior to code execution for a given period of time, to evaluate the instructions corruption effects. We choose an half-second accumulation time, to be pretty sure that with the experimentally measured error rate, alphas won't corrupt more than one bit per stored instruction. Moreover, half-second is a definitely a longer time with respect to execution time, which is in the order of milliseconds. Registers, as stated above, are not affected by this accumulation, as their values are continuously changed during execution and previous stored data are obsolete. Tab. 5.2 also reports the expected number of bits corrupted per accumulation time.

Table 5.3 reports the experimentally observed output errors for the ADD_Loop and 255_ADD codes execution with errors accumulation. As expected, 255_ADD is more likely to fail with respect to ADD_Loop. This is due to the high number of code

Table 5.3: *Experimentally measured errors per execution. Device has been exposed to radiation for half.second prior to code execution, so to let errors to accumulate*

	ADD_Loop	**255_ADD**
Errors detected	$19.03 \cdot 10^{-4}$	$40.31 \cdot 10^{-4}$
Timeouts	$5.56 \cdot 10^{-4}$	$8.00 \cdot 10^{-4}$

RAM bits needed to implement the sequential code.

Experimental results also stated that the expected value calculated and summarized by Tab. 5.2 and Fig. 5.1 are just a rough overestimation of the real device dynamic error rate. To deeply understand the reasons of the differences between expected and measured error rates, a low-level study of the different codes is necessary; the next paragraph gives the details of this analysis.

5.5.4 Results discussion

The first step to understand the different error rates, is a deep analysis of the assembly code. The *ADD A, #data* instruction is composed of two fields. The first is the 8 bit opcode 0x24 and the second holds the 8 bit data to be added. If the particle corrupts one of the latter 8 bits, a wrong addend will be selected, and so the final result will be wrong. When the opcode is corrupted, a new (wrong) instruction will be generated and the effects at the output will depend on the bit corrupted. Tab. 5.4 reports the instructions generated by one bit corruption. The radiation-induced possible effects on the computation are also reported. If bit 2 is corrupted, for instance, JB instruction will be generated and will likely lead to wrong program flow. On the contrary, if bit 5 is corrupted, no effect will be seen at the output. In fact, in this case, instead of *ADD A, #001* instruction the *INC A* will be fetched. This latter instruction simply increment the

Table 5.4: *Possible radiation induced effects on code RAM bits. ADD A, #data opcode is 0x24; for each opcode corrupted bit the wrong opcode generated, the wrong instruction it represents, and possible computation effects are reported*

Bit	Wrong opcode	Wrong instruction	Possible effects
0	0x25	ADD A, iram, addr	Wrong result/none
1	0x26	ADD A, @R0	Wrong result/none
2	0x20	JB bit addr, rel addr	Wrong program flow
3	0x2c	ADD A, R4	Wrong result/none
4	0x34	ADDC A, #data	Wrong result/none
5	0x04	INC A	None
6	0x64	XRL A, #data	Wrong result/none
7	0xA4	MUL AB	Wrong result

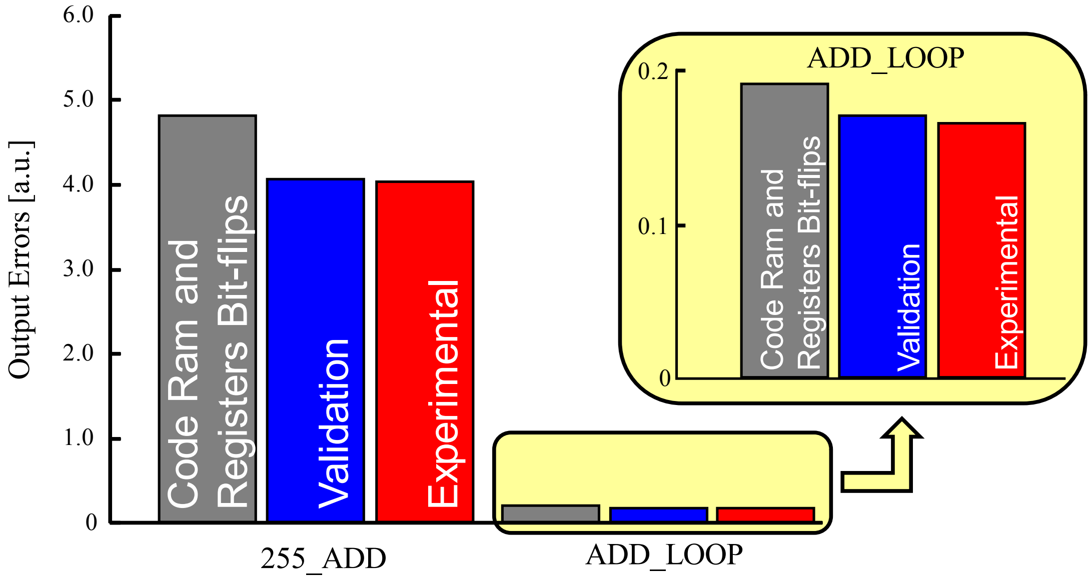

Figure 5.2: *Comparison between ADD_Loop and 255_ADD expected alpha induced bit-flips in the different memory resources used in the codes execution (code RAM and registers bit-flips, in grey), the experimentally observed output errors (experimental, in red), and the expected output error rate calculated taking the derating factors into account (validation, in blue)*

Accumulator value by one, which has exactly the same effect as the benchmark ADD instruction. If bit 6 is corrupted, the effect on the result depends on the value of the Accumulator. As we may assume the 8 bit #data to be correct, the new operation will be A xor 0x001. If the LSB of A is 1 it will be reset, leading to error but, if it is 0 it will be set, and the result will be the same as adding 1 to A. This suggests that the errors in the code execution will be lower than the values reported in table 5.2

The derating factor for 255_ADD is straightforward, as we know that 1/16 of the corrupted bits will generate INC instruction, and 1/32 a XRL with an even A. The resulting error prediction is $40.57 \cdot 10^{-4}$, still slightly higher than the experimentally measured one, reported in Tab. 5.3.

Register bits criticality is a delicate matter and needs further analysis, including fault injection experiments, which is one of the scopes of the following Chapter of this manuscript. However, as Tab. 5.3 suggests, in this particular case errors are dominated by code memory corruption. The contribution of register bits corruption to heighten the overall device radiation error rate is then minimal.

Fig. 5.2 shows how 255_ADD is definitely more sensitive to radiation with respect to the ADD_Loop code. Moreover, it’s easy to see how the experimentally observed number of output errors (in red) is definitely lower than the number of errors affecting the memory resources during the code execution (code RAM and registers bit-flips, in grey). With all the consideration stated in this paragraph, a derating factors for both the codes can be calculated, as not all the bit-flips will have effects on the code execution and will be propagated at the output. Applying the derating factor the expected number of errors affecting the memory resources is lowered (validation, in blue in Fig. 5.2) and fit well with the experimentally observed number of output errors. These considerations validate both our derating model and our low-cost test setup.

Finally, just few timeouts were detected. The probability of having bit 2 of one of the code instruction corrupted is $3.00 \cdot 10^{-4}$ slightly not enough to justify the $8.00 \cdot 10^{-4}$ timeouts per execution. Timeouts are then probably caused by microprocessor internal registers errors. The number of timeouts in the case of ADD_Loop is slightly lower. Other instruction as JUMP, MOV, SUB are involved in the sums calculations. Even if changing a bit in those instructions (JUMP in particular) will undoubtedly increase program flow jams, the probability of having their bits corrupted is much lower than 255_ADD, as only 144 bits are stored as code ram. Less errors were detected per execution attesting, once again, that internal register errors are masked or not critical during code execution.

5.6 Conclusions

Thanks to the described test flow, we have experimentally measure the sensitivity of the memory resources in a 8051-based SoC. Registers have a lower sensitivity to alphas than code RAM, due to the different structure of the flip-flops. To evaluate the overall device sensitivity, those data must be normalized to the number of bits used during a code execution and their effective criticality. Thanks to the results stemming from experiments on two different test benchmarks codes we demonstrate how code bits corruption may cause different effects at the device output.

Experimental data highlight that code memory corruption is a major concern. A higher number of instructions to be loaded causes a higher probability for code bits to be corrupted. It’s worth noticing that in both benchmark codes, the experimentally measured error rate is dominated by the code RAM errors.

This is a first step in the characterization of microprocessors radiation sensitivity starting from their memory resources cross section. Other tests will be carried out in order to calculate the sensitivity of different resources, and the derating factor of each. This will permit us to build an automatic tool that analyzes the assembly code to be loaded and gives an upper bound of its radiation sensitivity, thus possibly suggesting software design rules for lowering the device sensitivity while running the considered application.

Chapter 6

DFM Library Optimization Impact on Alpha Sensitivity

This Chapter presents and discusses the results of alpha Single Event Upset (SEU) tests on an embedded 8051 microprocessor core implemented in three different cell libraries. Each standard cell library is based on a different Design For Manufacturability (DFM) optimization strategy; our goal is to understand how these strategies may affect the device sensitivity to alpha-induced Soft Errors. The three implementations are tested exploiting advanced Design for Testability (DfT) methodologies and radiation experiments results are compared.

Our idea is to study and understand the impact of different levels of DFM layout optimizations, intended to increase product robustness and decrease yield losses, on the device radiation sensitivity.

We then focused our attention on alpha radiation tests applied to a microprocessor core embedded in a SoC and aim at determining the alpha-induced soft error rates when different implementation libraries are used, each one characterized by a different level of DFM rules implementation. As a case study, we describe the experiments performed on test vehicles manufactured by STMicroelectronics in a 90 nm technology and including an 8051 microprocessor. Radiation test results are provided *for each of the three DFM libraries used in layout synthesis.*

The Chapter is organized as follows: paragraph 6.1 introduces the Design For Manufacturing optimization and describes some hardening techniques, paragraph 6.2 introduces the different libraries on which the test chips were implemented, paragraph 6.3 provides a quick overview on radiation testing flows, paragraph 6.4 describes the experimental setup, paragraph 6.5 summarizes and discusses the experimental results, analyzing and proposing physical motivations for the results of the radiation experiments while paragraph 6.6 concludes the Chapter.

6.1 Design For Manufacturing

There are various strategies that can be used to increase devices dependability, both at layout and system or application level. Hardening techniques at high levels of abstraction include, for instance, Triple Modular Redundancy for logic cores and Error Correction Codes for memories [Nic01][Lim02]. At a lower level of abstraction, single devices may be hardened modifying their layout at physical and implementation level [Lim00].

When designing and hardening a layout to build a IC device in a given technology platform, Design Rule Manual (DRM) constraints must be strictly followed. A DRM is a set of mandatory layout basic rules a design has to comply with to be realizable in the fab. DRM includes, for instance, minimum space, minimum with, and other parameters constraints. Unfortunately, DRM compliance is only a necessary condition, but not a sufficient condition for a layout to be correctly implemented in silicon. In fact, imperfections and variations in a highly complex manufacturing process may cause different types of defects, both random and systematic, that compromise the correct silicon realization. These defects lead to yield losses. On one side, random defects have equal likelihood of occurrence and are mostly caused by the non-zero defect density in the manufacturing environment. On the other side, in nanometer fabrication processes with continuous shrinking of feature sizes, systematic failure are becoming more prominent [Hui04][Kru04][Mad04][Nig04]. Due to the drawing of sub-wavelength feature sizes in lithography processes, deformities occur in the printed layout. The gap between the feature size and wavelength is also increasing due to continuous shrinking of process technology. As a result, an increase in the occurrence of systematic defects is expected. The systematic failures are causing manufacturing excursion, high yield loss, and severe defects-per-million issues.

Several design rules and guidelines are followed to make the design manufacturing-friendly. Design For Manufacturing rules and guidelines consist of design rules and layout guidelines to ensure yield and manufacturability.

Design For Manufacturing (DFM) guidelines, or recommended rules, are an extension to the DRM, including more restrictive rules that may be applied opportunistically, aiming at increasing layout robustness in order to enhance the yield-learning process and shortening yield ramp-up. The design rules specify exact dimensions of width and spacing and are strictly followed in physical design. On the contrary, DFM guidelines are recommended layout practices. These guidelines are more restrictive but they are applied opportunistically. They are needed since it is not possible to cover or anticipate accurately all the process and layout interactions in the form of design rules [Kim07].

In general, standard cell design has mainly two conflicting needs: to optimize area occupation and to implement a robust layout with respect to process variability. A best compromise should be found on layout between conflicting DFM guidelines and different possible levels of application for the same layout recommendation. Indeed DFM requirements should be taken into account with the best trade-off decision versus all design requirements (e.g.: cell area, routability, timing preservation).

Due to constraints on layout geometry, die area and the ever decreasing window of time to market, complete information about process and fabrication defects is not known in advance. DFM optimization are applied when possible causes for systematic defects are identified but, to do so, the cell layout may be changed, which is time consuming, and, moreover, an increment in the cell area may be needed, which is silicon consuming. DFM optimizations may, then, increase manufacturing costs. However, the optimized cell has a higher robustness and reliability, and thus yield losses are reduced. DFM application is then a tradeoff between costs and benefits.

The effectiveness of overall hardening and mitigation strategies is traditionally proved through different testing and simulation campaigns. Radiation tests can be applied on dedicated test chips aiming at studying in detail the sensitivity of the different Intellectual Property cores (or simply IPs) composing a System-on-Chip (SoC). The integration of different functional modules supports the complexity of modern devices and usually requires specific implementation processes. This factor may affect the susceptibility levels towards Soft Errors (SEs) measured on chip arrays. Countermeasures may then be studied and eventually applied at the cores integration stage in addition to the ones introduced at lower levels of abstraction.

The selection of a SoC as a case of study is motivated by the need of observing different sensitivities caused by the specific topologies and related power supply distribution, which can be achieved in a realistic way only by testing the final SoC implementation and may not be observed testing stand-alone cores or cell arrays. Other radiation-induced effects (such as performance degradation) may indeed affect the correct interaction among SoC modules and can be only experimentally observed and measured in a complete system.

Efficient strategies are needed to collect data from Systems-on-Chip during radiation experiments and return precise information about the observed phenomena. Design for Testability / Diagnosability (DfT/D) circuitry added to the chips for manufacturing test purposes are here reused to ease data collection during radiation tests. The low-cost setup adopted during radiation tests has been described in details in the previous chapters, and it includes on-chip DfT/D structures and interfaces based on the IEEE 1500 Standard for Embedded Core Test (SECT) [IEE05], a suitable test board and a set of ad-hoc software procedures used for determining the sensitivity of different devices in a microprocessor-based system. The comparison of the gathered test results with the outcome of a massive simulative fault injection experiment enables the discrimination of cell-specific or location-dependent behaviors.

6.2 Test Vehicle Implementations

Radiation experiments were performed on an embedded microprocessor manufactured with three different ASIC standard cell libraries, each one enforcing a different grade of DFM rules. In this paragraph the characteristics of the three libraries are exposed. As explained before, we performed alpha radiation experiments on test vehicles manufactured by STMicroelectronics in a 90 nm technology, to investigate the radiation sensitivity of devices synthesized with standard cell libraries that implement three different levels of DFM optimization (Fig. 6.1).

Being that DFM rules are not orthogonal to each other, arbitrary choices were required at design stage to find out the most effective DFM trade-off for the considered circuit and technology. Usually, a criticality index is assigned to each guideline implemented during DFM optimization of layout. The criticality index is based on engineering knowledge of the process technology and previous technology nodes. The DFM optimization aims at individuating and increasing robustness of weak or critical layout configurations and is realized iterating automatic and manual steps, increasing or

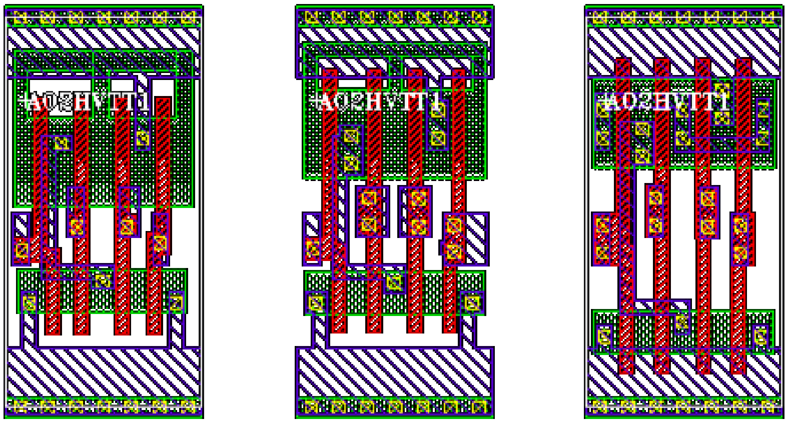

Figure 6.1: *Different DFM optimization levels applied to a cell.*

a) Library A (on the left) is the standard library

b) Library B (in the middle) has DFM optimization without increasing the cell area. This corresponds to the standard optimization approach.

c) Library C (on the right) is obtained with harder DFM optimization performed under extra area budget constrains

not the area of the library cells.

Objects of the testing campaign are three different realizations of a diagnosis-oriented SoC implemented using libraries that feature three levels of DFM maturity.

- **Library A** is the standard library in which no DFM optimizations are featured (Fig. 6.1a).

- **Library B** is an enhanced version of the previous one, optimized fixing all DFM critical configurations that are improvable using empty space in the cells (Fig. 6.1b). The layout of each single cell is strengthened, for instance, doubling contacts and vias, reducing weak layout configurations, and increasing application of DFM guidelines. It is important to note that all the modifications are done without increasing the area occupation of the cell. This corresponds to the standard DFM optimization approach.

- **Library C** is obtained with a stronger application of DFM guidelines under extra area budget constraints and it is likely to be the more dependable cell version. Electrical test results indicate some inflation on speed performances of this library, which is a direct consequence of some layout modification such as increased contact-to-gate distance (Fig. 6.1c).

To define the foot-print of the standard logic block, the layout synthesis was done at first with library C, which owns the largest cells. Then, layouts A and B were derived by substituting corresponding cells (with same logic functionality) in the predefined location and finally routing them suitably.

It is important to remark that DFM guidelines are typically applied to enhance layout robustness and reduce possible yield losses [Ait06][Kim07] but their effectiveness on lowering the error rate induced by device radiation has never been proven before and is one of the goals of this work.

6.3 Proposed Test Flow

For the current study, the testing methodology consists in forcing the processor to run a suitably developed benchmark code that is able to stimulate the different processor components and maximizes the effect of errors on observable points (i.e., output ports). The test operations, including code upload and activation and result compression, are managed through the test-support Infrastructure-IP beside the processor core.

The test application procedure requires the execution of the following operations:

- upload the benchmark code in a suitable memory area;
- activate its execution, i.e., letting the program run, stimulating the components under radiation;
- wait for the program run to complete, while a Multiple-Input Signature Register (MISR) connected to the processor output ports opportunely stores and compresses output data;
- retrieve the compressed signature after the benchmark code execution.

The characteristics of the MISR circuitry (e.g., length, primitive polynomial) were tuned to ensure a sufficiently low percentage of aliasing and cancellation due to multiple errors, and an acceptable silicon area overhead.

Fault simulation experiments, described later in this paragraph, helped at determining the criticality of the circuit resources to SEUs during the execution of a specific benchmark code, and for associating a faulty syndrome to specific SEU location in time and space.

6.4 Experimental Setup

The target SoC includes, as described in detail in Chapter 3, an 8-bit microprocessor (Intel 8051-compliant instruction set architecture), a 64Kx8 bit sized SRAM memory (perfectly symmetric and balanced bit cells) and a 16x16 parallel multiplier (ISCAS-85 C6288 benchmark [Brg85]). The memory block is partitioned so to be used as program and data memory by the processor; the multiplier is connected to the processor parallel I/O ports. By the manufacturing test point of view, it is possible to achieve high diagnosability for each of the cited components resorting to the following already cited DfT test structures. IEEE 1500 wrappers surrounding each core provide a common interface to the test logic. The external test controller interacts with the on-chip devices by sending high-level instructions. The test structures are then accessed through IEEE 1149.1 (JTAG, [IEE94]) Test Access Port (TAP), thus relying on only 5 signals for complete test control.

The DfT-intensive strategy permits at-speed testing (i.e., at the circuit nominal frequency) of integrated cores without the need of expensive ATEs or high speed connections. These structures were profitably reused for radiation experiments on the embedded microprocessor core, by devising a suitable software-based test flow. As described in the previous chapters, DfT enables high quality at-speed test and diagnosis while drastically lowering the cost of support equipment. Logic is stimulated at-speed preventing Single Event Transient underestimation and, moreover, Single Event Upsets are detected precisely through integrated self test structures. The integrated testing circuitry is exposed to radiation as the DUT and thus may be corrupted, however we demonstrated that, considering the testing circuitry limited area occupation and criticality, it is very unlikely for alphas to affect DfT structures [San08].

As explained in the previous chapters, the test interface board developed for manufacturing test and debug was reused for radiation testing. Voltage regulators supply the chip core and pads while a frequency modulator changes the operating frequency.

Since these parameters have a strong impact on device sensitivity to radiation, it is fundamental to gain data from experiments in the full range of variation of both supply voltage and working frequency [Bau05]. Our testing hardware ensures working frequency and supply voltage to be constant during the overall experiments campaign, again reducing the results experimental errors. The voltages may be changed with 0.02 V steps and frequency from 15 MHz to 200 MHz with 5 MHz steps.

Finally, a C++ software tool running on the host PC sends to the device the JTAG commands through its parallel port, collects and stores the test results, and controls the voltage regulators and the frequency modulator.

6.4.1 Radiation source

In order to determine the sensitivity to Alpha particles of the SoC implemented in the three different libraries, the set of radiation testing experiments was performed with an Americium alpha source available at DEI, Università di Padova, Italy. The accelerated radiation tests were performed with an Americium source emitting alpha particles. The shape of the ^{241}Am deposit is square, 35 mm wide, and the source activity is 250 kBq.

The Americium is deposited on 4 active strips covered with 2 μm of gold-palladium stuck on a stainless steel support which can be easily handled and placed on the chip under test. The half-time of ^{241}Am is 433 years, so the source can be modeled as a constant flux emitter. Alpha emission from the source is isotropic, therefore particles reach the die with different angles.

The Americium source is placed on a vertical shift platform that allows

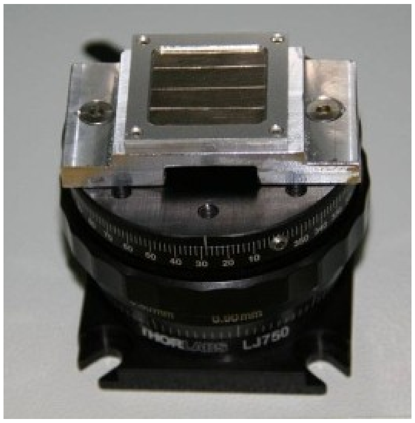

Figure 6.2: *Americium source placed on a calibrated vertical translation platform.*

calibrated vertical height changes, and eventually to position the source as close as possible to the DUT (Fig. 6.2). The distance between the silicon active area and the Americium source is only few millimeters, as the device is soldered on a daughter board without the need of any socket. Moreover, the ceramic package lid of the device can be easily removed, thus completely exposing the active area.

Our intention is to compare the radiation sensitivities of different devices, so experimental conditions should be very similar from test to test to reduce results experimental errors. Previous work [Bau07] demonstrated the significant effects of geometry and air absorption on accelerated alpha particle soft error rate tests. To minimize these effects we use the calibrated shift platform that ensures the distance of the radiation source from the die to be the same during our test campaigns and perform out experiments in a clean room, keeping air temperature (20°C) and humidity (70%) constant.

6.4.2 Fault simulation

We built a fault simulation system based on a logic simulator for determining the criticality of the circuit resources to SEUs during the execution of a specific benchmark code, and for associating each MISR signature to a specific SEU location and time of occurrence during the program run. The VHDL netlist of the whole SoC is available, thus permitting a complete and precise software simulation of errors effect and propagation. It is worth to note that having a register bit corrupted is not a sufficient condition for output failure. For instance, the corrupted register may not be used by the running application or the data it stores may be obsolete when the SEU occurs. Previous works show how some single-bit faults may not produce an error in a program's output. The Soft Error Rate of the device is then a strong function of the probability that a Soft Error will be observed at the outputs and the fraction of time a node or device is susceptible to upsets (named Architectural Vulnerability Factor and Timing Vulnerability Factor, respectively, by Mukherjee et al. and Seifert et al. [Muk03][Sei04]).

First of all, a stuck-at fault simulation is performed, executing the selected program and observing the MISR output signature. The addressed fault list includes the entire set of N circuit flip-flops (and data memory bits) used in the benchmark code execution. As it will be explained in the experimental results section, the effects of program memory corruption are not targeted in the current experiment. From this preliminary fault simulation step, we can sort out the flip-flops (and memory cells) that never affect the program execution. In fact, it is reasonable to believe that, if neither a stuck-at 0 nor a stuck-at 1 on a flip-flop provokes an output error during the program run, it is even less probable that a SEU on the same flip-flop will produce an error. For each of the F remaining ones ($F \leq N$), SEU fault simulations are performed.

Let the benchmark code run take T clock cycles. If we assume radiations to corrupt one bit per execution, we will have at most $F \cdot T$ different SEUs affecting the code execution. Therefore, $F \cdot T$ simulations are performed with the injection of one fault at a time, by toggling the value of flip-flop f at time t.

Each complete simulation provides a signature, which can be classified as follows:

- correct signature, when fault f at time t does not alter the expected signature (this happens, e.g., when a flip-flop is corrupted by alphas in a clock cycle preceding its rewriting and its content is then obsolete);

- wrong signature, identifying the specific SEU time and location;

- undetermined signature, if the injected bit flip alters the program execution flow in such a way that uninitialized memory elements bring their effect on the outputs.

This fault simulation methodology lets us determine the location and time of the flip-flop (or memory bit) corruption which leads to the specific wrong outputs, by providing a fault dictionary. Advanced fault-injection techniques, as the ones described in [Var00][Car02][Nic03], will be needed to take also SET into account. The VHDL description will, again, ease the understanding of errors propagation and effects.

A certain amount of aliasing is introduced when the same signature is associated to more than one fault injection run. Often the same signature is achieved for SEU injection on FF f at consecutive times t, t+1, t+2, …: this is due to the fact the contents of certain registers influence the processor behavior only when some specific instruction is executed, hence introducing some fault latency. In other cases, the same signature is obtained when injecting SEU effects on uncorrelated flip-flops: this less desirable situation is due to aliasing effects which can be overcome by increasing the observation of circuit outputs during the benchmark code run.

It is worth to note that the fault simulation results strongly depend on the selected benchmark code, and allow evaluating the resource criticality at architectural level. In other words, fault simulation allows determining the percentage of code execution time during which a SEU occurring at a specific location provokes a visible error. Moreover, fault simulation gives no indication on the different radiation sensitivities of devices built with different DFM libraries. In fact, the cells differ only at a physical level (i.e. layout design) while the logic schematic for all the tested devices is the same. Fault simulation results are then useful only if correlated with radiation experiments.

6.5 Experimental Results and Discussion

We performed both static and dynamic tests on our chips to study and compare their sensitivities to alpha radiation. Moreover, we correlated the experimental results to a fault simulation dictionary to try to understand and explain where DFM strategies are more effective in enhancing the device radiation sensitivity.

6.5.1 Static test

As a first characterization test we calculate the sensitivity to radiation of the 8051 internal registers under static conditions. We used an ad-hoc application that resets (or sets) each accessible register in the 8051 and enters an infinite loop. After a predefined time an interrupt subroutine is externally activated, which checks for corrupted bits, sending the number of detected mismatches to output ports that are downloaded through the JTAG.

We tested 5 chips for each library, and performed 100,000 static tests for each chip exposed to radiation. The measured sensitivity, reported in Fig. 6.3, shows that the microprocessor internal registers, built with standard cells implemented with different levels of DFM optimization, have different error rates, and library C appears to be definitely less sensitive to alpha radiation. This is a first interesting result demonstrating that applying higher level of DFM layout optimization to a IC device may improve its radiation robustness.

In particular, the application of DFM optimization without increasing the cell area (Library B) reduces the alpha sensitivity of 23% with respect to the standard cell (Library A), while the stronger application of the DFM optimization obtained increasing the cell area (Library C) reduces the sensitivity of 66% with respect to Library A. Higher maturity level of DFM, then, effectively enhances the device static robustness to radiation. The Library C optimization, in fact, reduces the static error rate of 56% with

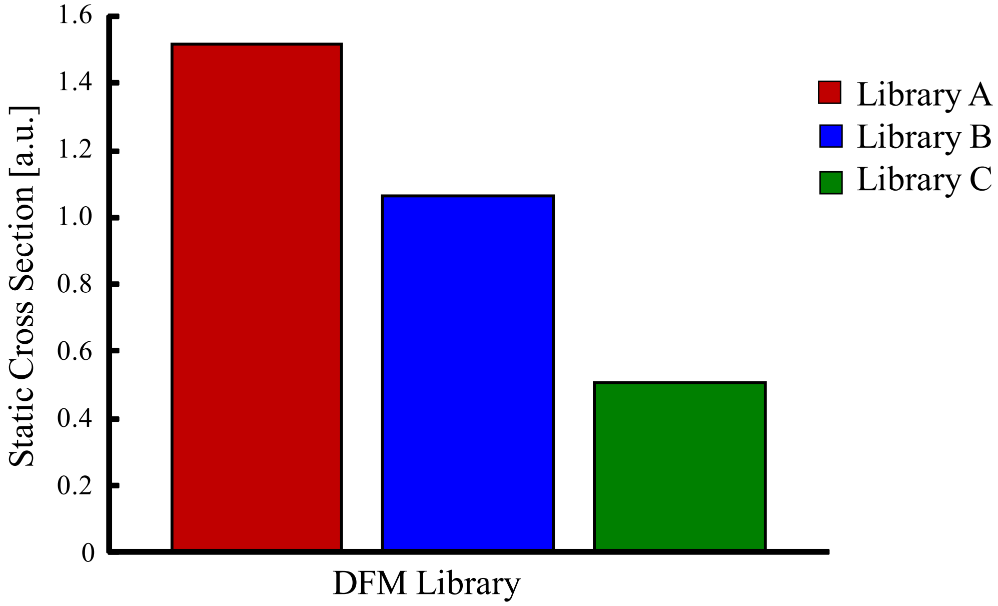

Figure 6.3: *Internal registers sensitivity to alpha radiation. For each different DFM library the number of errors detected in the 8051 internal registers normalized to exposed bits and unit time is reported.*

respect to the Library B application. DFM application, anyway, comes to a cost, as layout modifications are time consuming, and area increasing is silicon consuming.

6.5.2 Dynamic test

To understand the different DFM optimized microprocessors behavior when exposed to radiation, the static test is not sufficient. We used a particular benchmark code to be loaded in the radiation exposed microprocessor, so to measure its sensitivity when opportunely stimulated.

As a benchmark code to start studying the 8051 alphas sensitivity under operating conditions, we chose one of the test programs from an available manufacturing stuck-at test suite [Cor03]. Registers are not the only resource in a microprocessor that may cause wrong outputs. Combinational blocks, IO buffers, clock distribution lines and other parts may also be disturbed by radiation. It is then fundamental to design a benchmark code that stimulate a particular set of resources under dynamic conditions. The selected benchmark code is designed to stimulate a set of resources in the 8051 without working on external data memory, and is composed of move, jump and other instructions that are executed at 20 Mhz. Code outputs are continuously monitored by

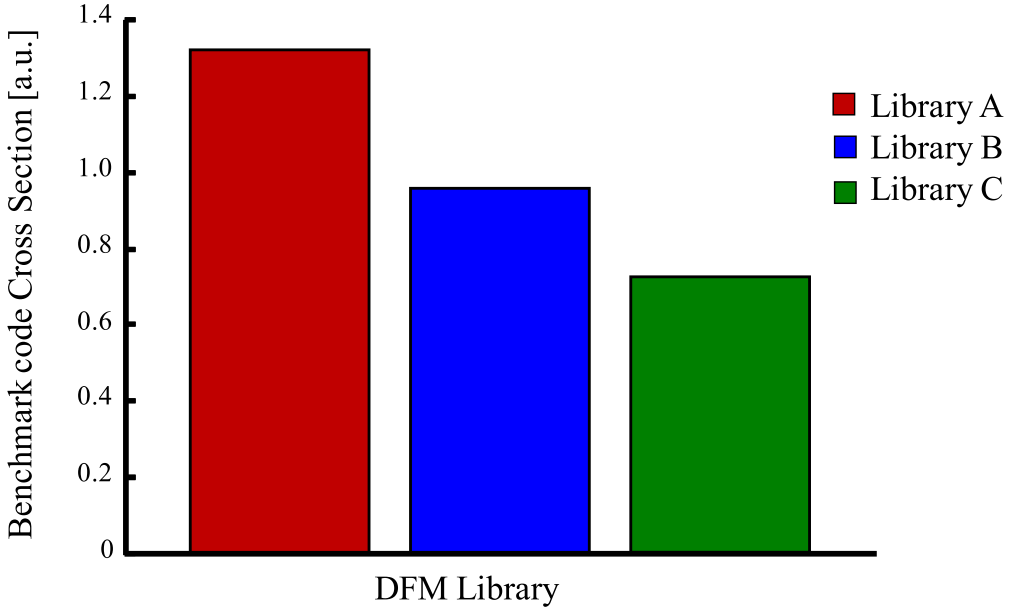

Figure 6.4: *Benchmark code cross section. For each DFM library the number of errors detected per benchmark code execution is reported. Library B and C have a similar error rate, which is definitely lower than the Library A one.*

the MISR circuitry, as described in the previous paragraph.

As the device is exposed to radiation, the benchmark code is then executed and the possible alpha-induced output error is sampled by the MISR circuitry. Once an error is detected, the faulty MISR signature is saved and a second code execution is performed to verify the possible occurrence of code memory errors that affect test results. If the latter execution gives correct MISR results, the test continues till the next error detection, as code memory may be assumed undamaged. On the contrary, when MISR results are identified again as wrong, code memory may have been corrupted by radiation and the obtained faulty signatures are discarded. The whole benchmark code is then reloaded in the microprocessor code memory and a new test is started.

We tested 5 chips for each library, and performed 200,000 benchmark code executions for each radiation exposed chip. Fig. 6.4 shows the radiation sensitivity of the 8051 for the different libraries. The number of errors detected by the MISR circuitry per benchmark code execution is reported for each different realization library.

Once again, we can conclude that library C, obtained with a stronger application of DFM guidelines under extra area budget constraints at cell level, is the more reliable to radiation. In particular, the application of DFM optimization without increasing the cell area (Library B) reduces the alpha sensitivity of 27% with respect to the standard cell (Library A), while the stronger application of the DFM optimization obtained increasing the cell area (Library C) reduces the sensitivity of 40% with respect to Library A. Standard cell area increase clearly ensures a higher resilience even to radiation, but this solution may introduce an area overhead at full-chip level which would increase the device manufacturing costs. Library B has also a lower error rate with respect to the standard library, while not increasing the area of the cells. Fig. 6.4 then shows how the increasing DFM maturity level ensures higher radiation reliability even under operating conditions. The Library C optimization, in fact, reduces the static error rate of 18% with respect to the Library B application.

6.5.3 Fault simulation results and discussion

Different types of flip-flops were used to build the device's internal registers, so their cells' structure, and therefore the DFM strategy applied, is not unique. As explained in paragraph 6.2, the DFM guidelines are not applied systematically, as it is required for DRM ground rules, but their application strongly depends on the layout of the cell to be hardened and specifically on the availability of unused space in the cell itself. To understand the DFM layout improvements whose contribution is stronger in increasing the device radiation reliability, we need first to detect the flip-flops that are most likely to fail and produce a MISR mismatch and then to study the differences between the three different layout implementations.

In order to achieve an estimation of the sensitivity of registers bits, and from

Table 6.1: *Cells' criticality in benchmark code execution. Percentages on columns show the distribution of library cells causing errors in the simulated fault injection experiment (independent on the implementation library) and in radiation tests (for each library).*

	Simulation	***Radiation Test***		
		Lib A	***Lib B***	***Lib C***
FD2TQHVTX1	52%	74.1%	71.9%	70.9%
FD4TQHVTX1	35%	24.1%	26.9%	27.1%
CTFD2TQHVTX4	5%	0.2%	0.2%	1.4%
FD2TQHVTX4	6%	1.1%	0.7%	0.4%
Others	2%	0.5%	0.3%	0.2%

Table 6.2: *Layout geometrical measurements done on all the transistors in one standard cell. The analysis is repeated for the corresponding versions of the cell present in the three libraries. The S/D region area and its percentage covered by Metal layer are also reported.*

	S/D region area *[μm^2]*	*Percentage of S/D region* *covered by Metal1*
FD4TQHVTX1_A	5.84	46%
FD4TQHVTX1_B	6.20	53%
FD4TQHVTX1_C	6.69	54%

there identify the flip-flop types that are most likely to fail during the benchmark code execution, we use the results of the fault simulation campaign. The fault simulation is based on the VHDL description of the overall SoC, and is thus of great help in understanding errors propagation and effects. Fault simulations campaigns give a faults dictionary: for each output wrong signature, a list of candidate cells whose corruption may have generated that signature is given. Thanks to the fault dictionary we have built, we are able to identify the specific register bit whose corruption is responsible of generating each faulty MISR signature obtained during radiation testing. This result is strictly associated to the executed benchmark code, but is independent on the layout implementation (and thus DFM optimizations) as each cell variant is designed with the same device logic schematic in the three libraries. The sensitivity of different flip-flop types in the overall 8051 radiation induced error rate is obtained correlating the resource criticality results determined through fault simulation with the experimentally determined SEU location statistic. From this elaboration, we determine that D flip-flop type FD2TQHVTX1 and FD4TQHVTX1 are the most sensitive ones: their contribution in the experimentally observed radiation induced output error rate is predominant with respect to all the other cells (Tab. 6.1).

As a first analysis to try to explain which DFM guideline ensures higher device reliability to radiation, we have studied the layout differences in the most critical cells structures among the three reference libraries. FD4TQHVTX1, in particular, is the cell in which DFM strategies found a more remarkable application. We performed layout geometrical measurements on all the transistors present in the FD4TQHVTX1 standard cell and in the corresponding version of the cell present in the three different libraries (Tab. 6.2). The first important difference we have noticed between the three different

Table 6.3: *Results of CC extraction performed at transistor level on the corresponding versions of the cell present in the three libraries. Capacitances here reported refer to NET vs GND. MOS capacitances are not included but their contribution can be considered (i.e. schematic design is the same).*

NET	LIB_A [fF]	LIB_B [fF]	LIB_C [fF]	% diff. (A to B)	% diff. (A to C)
SO	0.32	0.32	0.31	0%	-2%
net375	0.54	0.55	0.51	1%	-6%
CP	0.58	0.58	0.62	1%	8%
SD	0.69	0.71	0.66	4%	-4%
Q	0.69	0.78	0.78	13%	13%
D	0.83	0.80	0.78	-2%	-5%
DIN	0.83	0.73	0.88	-13%	5%
DIP	0.84	0.85	0.77	2%	-9%
TE	1.08	1.15	1.19	6%	10%
TI	1.14	1.11	1.05	-2%	-8%
TB	1.14	1.14	1.11	1%	-3%
M4	1.16	1.16	1.14	0%	-1%
M2	1.34	1.34	1.29	-1%	-4%
SDN	1.74	1.82	1.85	5%	7%
CPI	2.03	2.09	2.05	3%	1%
S1	2.05	2.12	2.20	3%	7%
M1	2.12	2.09	2.20	-2%	4%
vdd	3.18	3.25	3.45	2%	8%
CN	3.26	3.27	3.28	1%	1%
gnd	4.30	4.38	4.61	2%	7%

implementations is the increasing Source/Drain region in the Library B and Library C cells. This increases the probability for one particle to hit the active region but, on the other hand, enlarges also the node capacitance, making it more difficult for that particle to corrupt the cell. Moreover, the percentage of S/D area covered by Metal layer is higher in the DFM enhanced versions of the cells. This is due to the implementation of redundant active area-contact-metal interconnections applied following corresponding DFM guideline. Knowing that Metal layer is built with copper, we have simulated the range of Americium emitted alpha particles using SRIM [Zie08]. We observed that, even in the worst unlikely case of alphas hitting the device perpendicularly without losing energy during air interaction, some of impinging particles does not reach the active area below. So one of the possible explanations for the measured different sensitivity to alpha radiation is related to different usage of Metal layer that shields more the devices built with DFM enhanced libraries (i.e. Library B and C).

A second remarkable difference between the cell variants in the different libraries is the capacitance of the nodes and nets inside the cell. To estimate the different parasitic capacitances, we performed a transistor level CC extraction using STMicroelectronics sign-off extraction flow. The analysis considers only intra-cell interconnects (i.e., MOS capacitances are not included but their contribution can be considered negligible being that W and L parameters are unchanged in the three libraries) and the capacitances are related to each NET with respect to GND. As reported in Table III we found that, for some of the nets, the capacitance is higher in the Library B and C cells versions. We know that higher capacitance on the critical nodes and connected nets may increase radiation reliability [Dod03].

Unfortunately , the correlation between those and others parameters variation and alpha sensitivity is not straightforward, and is beyond the scope of this manuscript.

Further analyses are ongoing in order to focus on the nets that are on the critical path and connected to the flip-flop critical nodes (in Table 6.3 all the intra-cell nets are reported).

6.6 Conclusions

We have tested under radiation a set of devices designed and manufactured following different DFM approaches. The experimental results show how a higher level of DFM optimization of layout generally enhances the device resilience to alpha radiation. This proves to be related to the transistor layout, which increases nodes charge and different Metal layer areas. In addition, higher signal integrity is ensured by doubling techniques, and this may reduce radiation disturbs. Further tests and studies are going to be performed in order to provide a full understanding of the measured effect and correlate the different error rates with the physical cell differences in order to enhance the current DFM guidelines including specific radiation-aware layout recommendations.

To understand more deeply the reasons why the applied DFM techniques enhanced the robustness of the inspected cells, additional radiation experiments and simulation campaigns using SPICE or TCAD 3D are being planned.

The decision on which library to use when building a complex device is a hard-earned trade-off between costs, performance and, of course, reliability. Regarding the sensitivity to radiation, we demonstrated that the enforcement of yield-oriented DfM rules has an impact on reliability. This has to be taken into account when devising the mitigation strategy for a product, which depends on its requirements and on its mission environment, and can contribute to the general strategy based on additional software and hardware solutions (e.g., Error Correction Codes and Triple Modular Redundancy) to reduce the device error rate.

Chapter 7

TMR Effectiveness to Mitigate Errors Accumulation

Triple Module Redundancy is a powerful hardening technique heavily employed to enhance the radiation robustness of various devices. It was not possible to modify the System on Chip objective of the studies presented in the previous chapters layout to realize different TMR hardened chips. We then built an FPGA-based test setup to evaluate the efficiency of TMR when errors accumulation is concerned.

In this Chapter, then, we present an experimental analysis of alpha-induced soft errors in 90 nm low-end SRAM-based FPGAs. We first assess the relative sensitivity of the configuration memory bits controlling the different resources in the FPGA. We then study how SEU accumulation in the configuration memory impacts on the reliability of unhardened and hardened-by-design circuits. We analyze different hardening solutions comprising the use of a single voter, multiple voters, and feedback voters implemented with a commercial tool to understand TMR effectiveness in enhancing a general porpoise circuit reliability. Finally, we present an analytical model to predict the failure rate as function of the number of bit-flips in the configuration memory.

The main contribution of these studies is combining experimental measurements on low-end devices with the analytical analysis performed on the configuration data.

Experimental results are reported by first assessing the sensitivity of modern low-end SRAM-based FPGAs to alpha particles in both static (by measuring the sensitivity of the configuration memory bits controlling the different resources inside the FPGA, without running any applications) and dynamic tests (with an application executing in the device under test). Afterwards, we discuss the effectiveness of TMR techniques with different voting schemes in enhancing the system reliability, in the case where multiple errors are present in the configuration memory.

The Chapter is organized as follows: paragraph 6.1 gives an overview on FPGA devices, paragraph 6.2 presents the devices used in this work and illustrates the experimental setup; paragraph 6.3 describes the benchmark circuits used for the dynamic tests, and paragraph 6.4 shows the experimental results obtained irradiating the FPGA with alpha particles in both static and dynamic conditions (with and without hardening-by-design solutions), finally paragraph 6.5 provides an analytical model to interpret the obtained results and paragraph 6.6 concludes the Chapter.

7.1 SRAM based FPGA

RAM-based FPGAs are an attractive solution for many applications where short time-to-market, low-cost for low-production volumes, and in-the-field-programming ability are important issues. The versatility SRAM-based FPGAs offer comes from the adoption of a configuration memory (CFM) whose content defines the operations of the circuit the FPGA implements. It is therefore fundamental that the content of the configuration memory maintains the desired values during the FPGA operation.

One of the few major disadvantages of SRAM-based FPGAs is the sensitivity to ionizing radiation [Bel02][Swi04][Aus05]. Indeed, also at sea level, neutrons, originating from the interactions of cosmic rays with the atmosphere, and alpha particles, coming from radioactive contaminants in the package and solder material, may alter the content of the configuration memory through Single Event Upsets (SEUs). A change in the configuration memory can modify the implemented circuit, possibly leading to Single Event Functional Interruptions (SEFI) [Ful99][Ces02]. This is clearly unacceptable for safety-critical applications (especially those operating in radiation-harsh environments, such as space or nuclear power plants), but may be a serious issue also for mainstream applications, where the large diffusion of FPGA-based systems may lead to an unacceptable global failure rate (i.e., in a large population of chips), even if the single system has a low failure rate. Furthermore, the technological evolution is exacerbating this issue, since more scaled devices are usually more sensitive to ionizing particles.

Suitable hardening techniques are therefore needed to mitigate radiation effects in modern FPGAs. Rad-hard FPGAs are a solution only for specific applications due to their prohibitive cost and limited performance. Hardening-by-design techniques such as Triple Module Redundancy (TMR) [Lim01][Car01] are effective in preserving the design functionality when a SEU occurs. Scrubbing, i.e., the periodic refresh of the configuration memory, is another effective approach, especially when used in conjunction with TMR. Such techniques have been deeply investigated when applied to high-end devices (e.g., Xilinx's Virtex II/4/5), while few experimental data have been gathered for low-end devices (e.g., Xilinx's Spartan 3), which are likely to be the device of choice for mainstream applications (e.g., automotive) where cost reduction is a major concern. Furthermore, many works have focused on the impact of a single error in the configuration memory, neglecting the possibility of having multiple events due to several particles striking the device or to one single particle generating a Multiple Bit Upset. Though much less likely, this scenario may seriously challenge the effectiveness of traditional hardening techniques, such as Triple Module Redundancy.

7.2 Experimental Setup and Devices

For our experiments, we used a Spartan-3 XC3S200 designed by Xilinx in a 90 nm CMOS technology. The device features 4320 equivalent logic cells, 12 dedicated multipliers, 4 digital clock managers, 170 user I/O, 30 Kbits of distributed RAM, and 216 Kbits of Block RAM. The combination of low-cost and resource availability makes it suitable for many mainstream applications, such as in the automotive industry, where it is used to implement a variety of functions spanning from concentrating glue-logic on a single device to more complex data processing algorithms (e.g. digital audio filtering). In the case such devices are used in Electronic Control Units (ECU) managing critical vehicle functions (like steering or braking), it is mandatory to mitigate any effect that may prevent the FPGA from working as expected. Conversely, in the case such devices are used in not safety-critical functions, like for example in entertainment ECU, any effect that may prevent the FPGA from working correctly can reduce, even drastically, the quality of the service the ECU provides, and therefore it may have a dramatic impact on the user perception of the product quality. As a result, in both application scenarios, faults affecting the FPGA must be properly mitigated.

Our test-setup comprises a Device Under Test (DUT) board and a control board. The control board is equipped with a Xilinx Virtex-II Pro XC2VP30, whose Power PC is used to manage all the operations needed for performing both static and dynamic tests. It can configure and readback the DUT via JTAG, stimulate the DUT, and monitor the produced output. Radiation testing was performed in air using an Americium source emitting alpha particles with an energy of about 5.4 MeV and flux of $1.543 \cdot 10^4$ alphas s^{-1} within a solid angle of 2 sr. The half-time of ^{241}Am is very long, 433 years, so the source can be modeled as a constant flux emitter.

Prior to irradiation, the plastic package was etched through a nitric acid attack (Figure 7.1), leaving the die completely exposed.

7.3 Tested Configurations and Circuits

Initially, we performed static tests to estimate the alpha-induced error rate of the DUT configuration memory controlling the various resources inside the FPGA. The DUT was loaded with ad-hoc configurations and the Americium source was placed above the exposed die. The control board periodically scanned the DUT configuration memory searching for bit-flips. Afterwards, dynamic tests were carried out, comparing the DUT outputs with those coming from a golden unit not exposed to radiation. Readback and reconfiguration were performed either following a SEFI or after a given time elapsed from the previous readback. The corrupted bitstreams were post-processed using CILANTO [Vio05], to trace the bit-flips in the configuration memory back to the

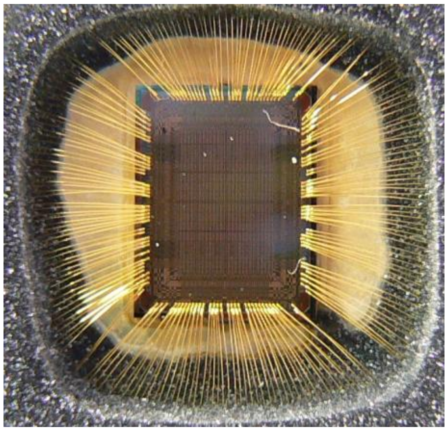

Figure 7.1: *Photograph of the etched FPGA exposed to the alpha flux*

controlled resources inside the FPGA.

One of the applications chosen for the dynamic tests was the PicoBlaze, a soft microcontroller (i.e., a microprocessor implemented using the FPGA fabric) freely available from Xilinx [Pic05]. The PicoBlaze structure is similar to the 8051, we decide to test the former one as its VHDL description, provided by Xilinx, is easier to be hardened through the X-TMR tool. Results similar to those presented here for the PicoBlaze were obtained also with other applications (e.g., a finite impulse response filter). The PicoBlaze consists of 16 8-bit registers, a 64-byte scratchpad RAM, a 1K-byte instruction ROM, and a 8-bit ALU. It occupies about the 5% of XC3S200 resources, performing 44 MIPS with a clock of 50 Mhz. The PicoBlaze was loaded with an assembly code implementing the functionality of an average moving filter. To

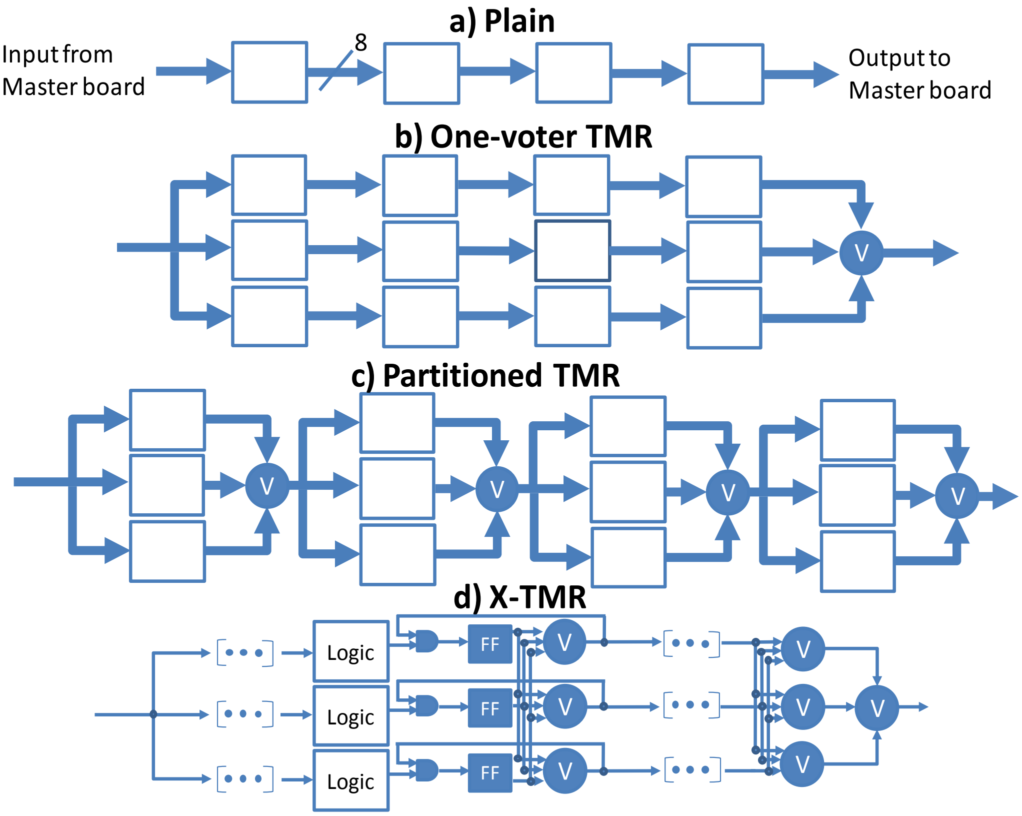

Figure 7.2: *Schematic of tested circuits with and without TMR hardening solutions. 7.2a is the plain version, without any TMR applied. In 7.2b the overall design is replicated and a voter is placed at the output. In 7.3c the design is divided into partitions and each partition is replicated and has a voter. In 7.4d hardening is performed by a commercial tool provided by Xilinx*

maximize resource usage and create an easy-to-partition design where to apply hardening techniques, we chained together four individual PicoBlaze units as shown in Fig. 2a. All the PicoBlaze instances perform the same task (a simple averaging filter); the outputs of a chain element are connected to the inputs of the following stage. After assessing the sensitivity of the unhardened circuit to alpha particles, we applied different mitigation schemes based on TMR.

In particular we adopted the following three solutions:

- **One-voter TMR:** the design is replicated three times and a majority voter is placed at the circuit output performing a bit-by-bit voting (Figure 2b).

- **Partitioned TMR:** the unhardened design is divided into different partitions. Each partition is replicated three times and a majority voter is adopted on each partition's output (Figure 2c).

- **X-TMR:** hardening is performed using a commercial tool provided by Xilinx [Tmr04]. Feedback voters are inserted to keep the state of FSM synchronized across each replica of the circuit (Figure 2d).

All the circuits were clocked at 10 MHz during our tests, thus minimizing errors due to Single Event Transients (SET).

7.4 Experimental Results

We first performed a static test to characterize the device resources sensitivity to alphas. The first result obtained is that the 0 to 1 bit-flip (0→1), which is the corruption of a bit set to 0 into a 1 and the 1 to 0 (1→0) bit-flip have different probability of occurrence. The data collected during the static tests are presented in Tab. 7.1, where the cross section for each resource is normalized to the 1→0 LUT bit-flip (details about the various FPGA resources may be found in [Ste06_1] and [Ste06_2]). As shown, LUTs are the most sensitive resource to alpha particles. In addition, for all resources the probability of 0→1 and 1→0 upsets are different, possibly due to asymmetric physical layout and/or asymmetric capacitive load. These data are particularly important, since they allow a designer to predict the soft error sensitivity of a given circuit implemented in the FPGA, knowing only the used resources as we will show later.

Table 7.1: *Alpha sensitivity of the configuration memory controlling different FPGA resources and BRAM (normalized to LUT 1 to 0 bit-flip cross section)*

FPGA resource	*Configuration bits [#]*	*Normalized cross section of 1 to 0 transitions*	*Normalized cross section of 0 to 1 transitions*
LUTs	61,440	1.00	1.29
MUXs	61,440	0.25	0.82
Slice Conf	61,440	0.61	1.08
Decoded PIP	245,760	0.38	0.90
Non-dec PIP	153,600	0.46	0.81
User memory	225,024	0.84	0.93

Table 7.2: *Resource occupied by the tested designs*

Design	*LUT*	*MUX*	*CFG*	*DPIP*	*NPIP*	*# Voters*
Unhardened PicoBlaze chain	9,488	3,276	1,699	8,570	4,759	0
One-voter TMR PicoBlaze chain	29,232	9,878	5,317	27,301	15,428	8
Partitioned TMR PicoBlaze chain	39,968	10,051	5,584	28,330	16,089	32
X-TMR Picoblaze chain	34,800	10,643	6,956	36,283	23,292	344

Concerning the dynamic tests, the resource usage of the designs exposed to alpha particles is summarized in Tab. 7.2, while Tab. 7.3 and Fig. 7.3 present the experimental results. Qualitatively similar results were obtained also with other circuits (e.g., a Finite Impulse Filter).

As our data show, TMR techniques are very effective in mitigating soft-errors when a single or just a few SEUs occur in the configuration memory, but some of them may completely lose their effectiveness when SEU accumulation occurs. For instance, the failure rate of the one-voter TMR version is worse than that of the plain one with 16 errors in the configuration memory. Partitioned TMR can offer increased robustness, depending on the number of partitions in the design and the circuit itself. Yet, for large error accumulation, the improvement may be only marginal. The feedback voters

Table 7.3: *Alpha source experimental results for dynamic circuits*

Design	*SEFI/min reconfiguring after 5 bit-flips*	*SEFI/min reconfiguring after 10 bit-flips*	*SEFI/min reconfiguring after 16 bit-flips*	*SEFI/min reconfiguring after SEFI*
Unhardened PicoBlaze chain	0.35	0.87	0.88	1.16
One-voter TMR PicoBlaze chain	0.18	0.65	0.90	1.43
Partitioned TMR PicoBlaze chain	0.06	0.22	0.36	0.91
X-TMR Picoblaze chain	0.03	0.14	0.17	0.51

introduced by X-TMR can further improve the application reliability, effectively creating a large number of partitions in the design.

7.5 Analytical Model

Starting from the results obtain during radiation experiments, we want to build a model to describe the different TMR strategies effectiveness in enhancing a general circuit reliability to radiation. This model could then be applied also to a general System on Chip, as the one described in the previous chapters, to predict how the application of TMR affects the radiation induced error rate of the device.

Previous work [Ste05] showed that assuming only a single bit-flip in the configuration memory a worst-case estimation of the sensitivity of a given circuit is given by the number of used bits divided by the total number of configuration memory bits. From the collected static data and from the analysis of the used resources we developed a refined model to predict the failure probability in presence of multiple SEUs in the configuration memory. The model can be summarized by Equation 7.1, where $n_{1,resource}$ ($n_{0,resource}$) is the number of configuration memory bits set to 1 (0) relative to a given resource in the slices used by the circuit; $w_{1,resource}$ ($w_{0,resource}$) is the probability that a 1→0 (0→1) transition in the configuration memory bits controlling resource leads to a functional interruption; $\sigma_{resource,1\rightarrow 0}$ ($\sigma_{resource,0\rightarrow 1}$) is the experimental upset cross section of the configuration memory bits for 1→0 (0→1) transitions controlling resource;

$$\sigma_{design} = \sum_{all-resources} n_{1,resource} \cdot w_{1,resource} \cdot \sigma_{resource,1->0} + n_{0,resource} \cdot w_{0,resource} \cdot \sigma_{resource,0->1} [\cdot d_{1,resource}]$$

Equation 7.1: *Analytical model to estimate the sensitivity of unhardened circuits*

$d_{1,resource}$ is the density of 1's and must be included for those resources where the probability that an added resource interferes with the circuit functionality increases with the number of resources of that type already present.

For instance bit-flips in a LUT used to implement a logic function inside an FPGA will result in an error at the outputs regardless of being 0→1 or 1→0 transitions, obviously assuming that the workload uses that LUT, hence $w_{1,LUT}$ is equal to 1. Conversely, bit flips in the configuration memory controlling non-decoded PIPs will surely impact on the application in the case of 1→0 transitions, since those correspond to the removal of existing connections; but they may or may not have an impact in the case of 0→1 transitions, since those correspond to the addition of a path which may o may not interfere with existing connections. Of course, the larger the number of interconnections, the higher the probability that an added interconnection interferes with the application routing. This turns into the necessity of including d1,non-decoded PIPs in the calculation. Equation 7.1 states that the dynamic sensitivity of an FPGA is less than its static sensitivity. In other words, not all the bit-flips in the configuration memory lead to an error at the outputs, depending on different parameters.

Equation 7.1 can be used to compare the sensitivities of different circuits implemented in the FPGA. We compared a broad range of combinational and sequential designs (including the PicoBlaze application described in this article), both experimentally and with our analytical model, and found an agreement ranging from 5 to 10% between measurements and analytical predictions.

We developed a model to obtain the failure probability of the hardened designs as a function of the number of bit-flips in the configuration memory, starting from the radiation sensitivity of the plain version. For this purpose we used the following (simplified) assumptions:

i. the configuration memory of a plain circuit is made of sensitive (upsets in these bits lead to an error in the output at least for certain inputs) and insensitive (no errors can be caused by upsets in these bits) bits

$$
\begin{aligned}
&W_{plain}(e) = W_{plain}(e-1)\cdot(m-s)\\
&SEFI_{plain}(e) = W_{plain}(e-1)\cdot s + SEFI_{plain}(e-1)\cdot m\\
&W_{one-voter}(e) = W_{one-voter}(e-1)\cdot(m-t\cdot s)\\
&FR_{1one-voter}(e) = W_{one-voter}(e-1)\cdot t\cdot s + FR_{1one-voter}(e-1)\cdot(m-(t-1)\cdot s)\\
&SEFI_{one-voter}(e) = FR_{1one-voter}(e-1)\cdot(t-1)\cdot s + SEFI_{one-voter}(e-1)\cdot m\\
&W_{part}(e) = W_{part}(e-1)\cdot(m-t\cdot s)\\
&FR_{i\,part}(e) = FR_{i\,part}(e-1)\cdot(m-t\cdot s+i\cdot s/p) + FR_{i-1\,part}(e-1)\cdot(p-i+1)\cdot t\cdot s/p \qquad i=1,2,\ldots p\\
&SEFI_{part}(e) = FR_{i\,part}(e-1)\cdot i\cdot(t-1)\cdot s/p + SEFI_{part}(e-1)\cdot m
\end{aligned}
$$

Equation 7.2: *Analytical model to estimate hardened-by-design circuit sensitivity as a function of the number of errors in the configuration memory.*

ii. if the number of sensitive bits in the unmitigated version is s out of a total of m configuration memory bits, it is t·s in the triplicated ones, where t (overhead factor) is slightly larger than 3;

iii. triplicated versions can fail only if there are at least two bit-flips;

iv. design partitions have the same number of sensitive bits s/p for the plain version and each TMR domain.

We must remark that these hypotheses are only approximate: TMR can fail even after a single bit-flip due to multiple effects, partitions lengths may be uneven, and the sensitivity of the different bits is not the same, as shown in the previous paragraph. Nevertheless, even with these simplifying assumptions we can obtain an adequate explanation of our experimental results. When TMR hardening techniques are used, triplication and design partitioning strongly impact the failure probability. This can be calculated with the iterative equations 7.2, where e is the number of bit-flips in the configuration memory, m is the total number of configuration memory bits, and p the number of equal partitions in which a triplicated design is divided. Since me is the total number of possible combinations in which e configuration bits may be upset, W(e)/m^e is the probability that a design correctly works with e errors in the configuration memory, SEFI(e)/m^e is the probability of a functional interruption with e errors in the configuration memory, and FRi(e)/m^e is the probability that a replica fails in one of the i partitions of the triplicated design (but no errors appear at the output). In other words, Equations 7.2 state that:

i. an unmitigated version can fail whenever a sensitive bit is upset;

ii. one-voter TMR fails if two sensitive bits belonging to two different replicas are upset;

iii. partitioned TMR fails if two sensitive bits belonging to two different replicas of the same design partition are upset.

The derivation is quite straightforward. For instance, the probability that an unmitigated version correctly works with one error in the CFM is equal to the probability that a non critical bit has been affected, i.e. m-s/m. Then, the probability of correct operation after i errors in the CFM, is given by the probability that it works with

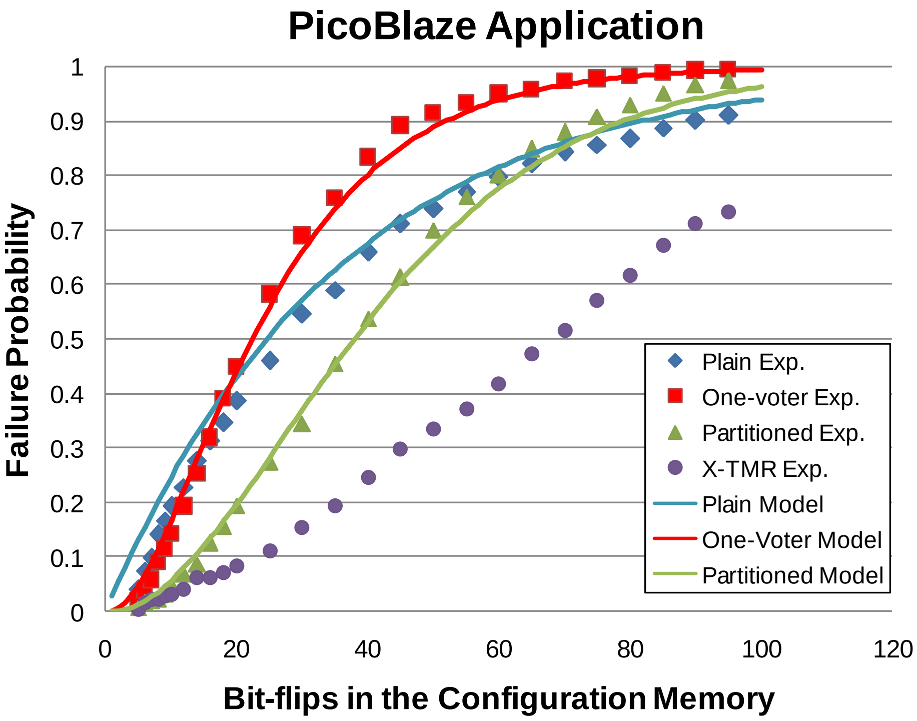

Figure 7.3: *Comparison between experimental data and model*

i-1 errors, multiplied by (m-s)/m. With one-voter TMR, one has to consider separate probabilities for the three replicas of the circuit: when two replicas fail the whole circuit fails (within our simplified assumptions). Partitioned TMR can be analyzed in a similar manner, assuming a failure occurs when the same design partition fails in two replicas.

Our model correctly reproduces the observed experimental results. For instance, Fig. 7.3 shows the failure probability as a function of the number of bit-flips in the configuration memory for the PicoBlaze application we presented before, as measured experimentally and as deduced from our model. The model parameters were m=1,000,000 (the number of configuration bits in the whole FPGA under test), s=27,792 the number of sensitive bits (see Table 7.2), p=4 (the number of equal design partitions), t=3.23 (the overhead factor for the triplicated versions). At the moment, we only show the experimental data for the X-TMR version, the analytical model is more complex and will be developed during future work.

Interestingly enough, for small (the number depends on the implemented application) accumulations of bit-flips in the configuration memory triplication reduces the failure rate of the examined circuits. Yet, as the number of errors which are permitted to accumulate in the configuration memory grows, one-voter TMR loses its effectiveness with respect to the unmitigated version. Partitioned TMR helps to reduce the failure probability also with a larger numbers of bit-flips as compared to one-voter TMR. The maximum number of errors in the configuration memory for which triplication is effective depends on the overhead factor, the number of partitions in the design, and the extent of each partition.

7.6 Conclusions

TMR is a powerful widely used tool to mitigate radiation effects. We have showed an experimental study on the alpha-sensitivity of low-end SRAM-based FPGAs, focusing on the occurrence of multiple SEUs or MBUs in the configuration memory. We measured the alpha-sensitivity of the configuration memory cells controlling the different resources an SRAM-based FPGA embeds, so to refine the characterization of our device. We performed dynamic tests of a complex circuit with and without hardening solutions based on TMR, measuring the rate of functional interrupts during exposure. This permitted us to understand how the TMR effectively enhances the device radiation resilience to radiation. The robustness of each design was discussed as a function of the voting scheme and the number of SEUs accumulated in the FPGA configuration memory. We also developed an analytical model to predict the failure probability of a circuit hardened with TMR in the presence of multiple errors in the configuration memory. This model will be very useful to predict the effectiveness of the TMR strategy applied to different kind of circuits and devices, as the System on Chip described in the previous chapters.

Conclusions and Future Works

Radiation is an issue for both space and terrestrial electronic applications. Large scale terrestrial electronic devices are affected by radiation, and with technology evolution an increment in the number of radiation induced errors is expected. Moreover, the ongoing increasing occurrence of Multiple Bit Upsets will make useless most of the today widely use hardening techniques as Error Correction Codes or Triple Module Redundancy, as more bits inside a single word or in different domains may be corrupted. Not only, also Single Event Transients will grow of importance and danger. With the newest technology nodes, SETs cannot be underestimated.

In this scenario, the characterization of electronic devices becomes an important step in the qualification process, so to understand if the device is employable in fields that traditionally demand high reliability, as the automotive and the biomedical ones. There are various testing strategies that can be applied to radiation experiments, we believe that, to analyze the radiation effects on real-world devices, experiments should be performed on the complex ICs such as System on Chips final implementation. In fact, even if tests on different stand alone cores arrays are easier to perform, there are various radiation induced effects than can be observed only in the implemented chip, as performances degradations, for instance. Moreover, testing the final SoC implementation will permit to understand how the different cores corruption affect the overall system functionality.

We proposed and demonstrated the effectiveness of low-cost radiation test approaches based on the reuse of on-chip DfT logic. We described a test flow that allowed us to experimentally measure the sensitivity to alphas and neutrons of an embedded SRAM core, an embedded logic core, and a microprocessor core inside a SoC. DfT structures were added for manufacturing qualification porpoises, our idea is to reuse this built-in circuitry to measure the SoC sensitivity to different impinging particles and to understand how the different cores corruption affects the system functionality. Thanks to the built-in structures, which provide precise information about failures, the test can be performed at operating condition, thus giving a realistic idea of the SoC behaviour when exposed to radiation. The IEEE 1500 wrappers ease cores accessibility and the JTAG interface permits just low speed connection between the DUT and the controlling hardware. This solution is very effective in easing radiation tests, as it avoids the use of expensive ATE or high speed connections to monitor the DUT executions.

We performed various radiation experiments on the different cores available in the SoC to validate our testing strategy and to gain precise information on the overall system behaviour when exposed to both neutrons and alpha particles. SRAM core tests pointed out the robustness of our solution, in particular the pBIST gave information about time and location of the radiation induced bit-flips which is fundamental to detect Multiple Bit Upsets.

The experiments on the embedded microprocessor pointed out that internal registers have a higher sensitivity to alphas than code RAM, due to the different structure of the flip-flops. We normalized those data to the number of bits used during a code execution and calculated their effective criticality, so to evaluate the overall device sensitivity. Thanks to the results stemming from experiments on two different test benchmarks codes we’ve demonstrated how code bits corruption may cause different effects at the device output. Experimental data highlight that code memory corruption is a major concern. A higher number of instructions to be loaded causes a higher probability for code bits to be corrupted. This was just a first step in the characterization of microprocessors radiation sensitivity starting from their memory resources cross section. Other tests will be carried out in order to calculate the sensitivity of different resources, and the derating factor of each. This will permit us to build an automatic tool that analyzes the assembly code to be loaded and gives an upper bound of its radiation sensitivity, thus possibly suggesting software design rules for lowering the device sensitivity while running the considered application.

We have also tried to study the impact of different hardening strategies applied to the microprocessor. We have tested under radiation a set of devices designed and manufactured following different Design For Manufacturing approaches. The experimental results show how a higher level of DFM optimization of layout generally enhances the device resilience to alpha radiation. This proves to be related to the transistor layout modifications, which increases nodes charge and different Metal layer areas. In addition, higher signal integrity is ensured by doubling techniques, thus reducing radiation disturbs. Further tests and studies are going to be performed in order to provide a full understanding of the measured effect and correlate the different error rates with the physical cell differences in order to enhance the current DFM guidelines including specific radiation-aware layout recommendations. The decision on which library to use when building a complex device is a hard-earned trade-off between costs, performance and, of course, reliability. Regarding the sensitivity to radiation, we demonstrated that the enforcement of yield-oriented DFM rules has an impact on reliability. This has to be taken into account when devising the mitigation strategy for a product, which depends on its requirements and on its mission environment, and can contribute to the general strategy based on additional software and hardware solutions to reduce the device error rate. We have also study and propose a model to predict the TMR efficiency when errors accumulation is concern.

Finally, on field high altitudes experiments are going to be performed. Many works presented in IOLTS 2009 [Hub09] and RADECS 2009 [Hei09] attested how accelerated radiation tests are very useful, but their results must be correlated to the exposure to the natural particles flux. Heijmen, in particular, presented data that demonstrate how high altitude and underground real-time measured neutron and alpha induced Soft Error Rate of embedded SRAM core somehow differs from the accelerated test extrapolated ones. In particular, radiation tests that use radioactive sources as alpha emitters seem less accurate than neutrons experiments extrapolations. We believe that our strategy can be fruitfully applied also to real-time high altitude and underground tests. The monolithic shape of the test board makes it easy to be placed in any support, and its robustness has already be proven during the massive test campaigns objective of this manuscript. Moreover, only low-speed communications are needed between the test board and the controlling hardware, making experiments remote controls easier.

Bibliography

[Ada91] L. Adams, E.J. Daly, R. Harboe-Sørensen, A.G. Holmes-Siedle, A.K. Ward, and R.A. Bell, “Measurements of SEU and Total Dose in Geostationary Orbit Under Normal and Solar Flare Conditions”, *IEEE Trans. Nucl. Sci.*, Dec. 1991, vol. 38, no. 6, pp. 1686-1692

[Ait06] R. Aitken, “DFM Metrics for Standard Cells”, *in proc. International Symposium on Quality Electronic Design (ISQED’06)*

[App03] D. Appello, P. Bernardi, A. Fudoli, M. Rebaudengo, M. Sonza Reorda, V. Tancorre, and M. Violante, “Exploiting Programmable BIST for the Diagnosis of Embedded Memory Cores”, *in proc. of International Test Conference 2003*, pp. 379-385

[Aus05] Austin Lesea, Saar Drimer, Joseph Fabula, Carl Carmichael, and Peter Alfke, “The Rosetta Experiment: Atmospheric Soft Error Rate Testing in Differing Technology FPGAs”, *IEEE Transactions on Device and Materials Reliability*, Vol. 5, Number 3, September, 2005

[Axn86] C.L. Axness, H.T. Weaver, J.S. Fu, R. Koga, and W.A. Kolasinski, “Mechanisms leading to Single Event Upset”, *IEEE Trans. Nucl. Sci.*, Dec. 1986, vol. 33, pp. 1577-1580

[Bar97] P.H. Bardell, W.H. McAnney, and J. Savir, “Built-In Test for VLSI: Pseudorandom Techniques”, *Wiley Interscience*, 1987

[Bau02] R. Baumann, “The Impact of Technology Scaling on Soft Error Rate Performance and limits to the Efficacy of Error Correction”, *in proc. of Int. Electron. Devices Meeting (IEDM) Tech. Dig.*, San Francisco, CA, Dec. 2002, pp. 329-332

[Bau05] R. Baumann, “Radiation-Induced Soft Errors in Advanced Semiconductor Technologies”, *IEEE Trans. Nucl. Sci. Device and Material Reliability*, Vol. 5, Sept. 2005, pp. 305-316

[Bau07] R. Baumann and D. Radaelli, “Determination of Geometry and Absorption Effects and Their Impact on the Accuracy of Alpha Particle Soft Error Rate Extrapolations”, *IEEE Trans. Nucl. Sci.*, vol. 54, no. 6, Dec. 2007, pp. 2141-2148

[Bel02] M. Bellato, M. Ceschia, M. Menichelli, A. Papi, J. Wyss and A. Paccagnella, “Ion Beam Testing of SRAM-based FPGA’s”, *IEEE Radiation Effects Data Workshop*, July 2002

[Bel04] M. Bellato, P. Bernardi, D. Bortolato, A. Candelori, M. Ceschia, A. Paccagnella, M. Rebaudengo, M. Sonza Reorda, M. Violante, and P. Zambolin, “Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA”, *in proc. of Design, Automation and Test in Europe*, 2004, pp. 188-193

[Bel82] C. Bellon, A. Liothin, S. Sadier, G. Saucier, R. Velazco, F. Grillot, and M. Issenman, “Automatic Generation of Microprocessor Test Programs”, *in proc. of 19th Design Automation Conference*, 1982, pp. 566-572

[Ber04] P. Bernardi, M. Rebaudengo, and M. Sonza Reorda, “Using Infrastructure IP to Support SW-based Self-Test of Processor Cores”, *in proc. of IEEE International Workshop on Microprocessor Test and Verification*, 2004, pp. 22-27

[Ber05_1] P. Bernardi, C. Masera, F. Quaglio, M. Sonza Reorda, “Testing logic cores using a BIST P1500 compliant approach: a case of study”, *in proc. of IEEE Design Automation and Test in Europe Conference*, 2005, pp. 228-233

[Ber05_2] P. Bernardi, M. Grosso, M. Rebaudengo, M. Sonza Reorda, “Exploiting an I-IP for both test and silicon debug of microprocessor cores”, *in proc. IEEE International Workshop on Microprocessor Test and Verification*, 2005, pp. 55-62

[Ber09] P.Bernardi, M. Grosso, P. Rech, M. Sonza Reorda, D. Appello, S. Gerardin, and A. Paccagnella, “DfT Reuse for Low-Cost Radiation Testing of SoCs: a case study”, *in proc. IEEE VLSI Test Symposium 2009*, Santa Cruz, California, pp. 276-281

[Bin75] D. Binderm E.C. Smith, and A.B. Holman, *IEEE Trans. Nucl. Sci.*, NS-22, 2675 (1975)

[Brg85] F. Brglez and H. Fujiwara, “A neutral netlist of 10 combinatorial benchmark circuits and a target translator in FORTRAN”, *in proc. Int. Symposium on Circuits and Systems*, 1985, 663-698

[Buc97] S. Bucner, M. Baze, D. Brown, D. McMorrow, and J. Melinger, “Comparison of Error Rates in Combinations and Sequential Logic”, *IEEE Trans. Nucl. Sci.*, Dec. 1997, vol. 44, no. 6, pp. 2209-2216

[Car01] C. Carmichael, “Triple Module Redundancy Design Techniques for Virtex FPGAs”, *Xilinx Application Note XAPP197*, Nov. 2001

[Car02] G.C. Cardarilli, F. Kaddour, A. Leandri, M. Ottavi, S. Pontarelli, and R. Velazco, “Bit flip injection in processor-based architectures: a case study”, *in proc. of the 8th IEEE International On-Line Testing Workshop*, 2002

[Ces02] M. Ceschia, A. Paccagnella, S.-C. Lee, C. Wan, M. Bellato, M. Menichelli, A. Papi, A. Kaminski and J. Wyss, “Ion Beam Testing of ALTERA APEX FPGAs”, *NSREC 2002 Radiation Effects Data Workshop Record*, Phoenix, AZ, USA, July 2002

[Cho97] R. Chou, K. Saluja, and V. Agrawal, *“Scheduling Tests for VLSI systems under power constraints”, IEEE Trans. VLSI Systems*, vol. 5, no. 2, Sept. 1997, pp. 175-185

[Coc94] B.F Cockburn, “Tutorial on semiconductor memory testing”, *Journal of Electronic Testing: Theory and Application*, vol. 5, no.4, Nov. 1994, pp. 321-336

[Cor03] F. Corno, G. Cumani, M. Sonza Reorda, and M. Squillero, "Fully automatic test program generation for microprocessor cores", *in proc. Design, Automation and Test in Europe Conference and Exhibition*, 2003, pp. 1006-1011

[Cro97] J.W. Cronin, T.K. Gaisser, and S.P. Swordy, “Cosmic Rays at the Energy Frontier”, *Scientific American*, January 1997

[Dag01] I.A. Daglis, “Space Storms and Space Weather Hazards”, vol. 38, Chapter 3, *Kluwer Academic Publishers*, 2001

[Det97] C, Detcheverry, C. Dachs, E. Lorfèvre, C. Sudre, G. Bruguier, J.M. Palau, J. Gasiot, and R. Ecoffet, “SEU critical charge and sensitive area in a submicron CMOS technology”, *IEEE Trans. Nucl. Sci.*, Dec. 1997, vol. 44, pp. 2266-2273

[Dod02] P. E. Dodd, M. R. Shaneyfelt, J. R. Schwank, and G. L. Hash “Neutron-induced soft errors, latchup, and comparison of SER test methods for SRAM technologies”, *in proc. Int. Electron Device Meeting*, 2002, pp. 333-336

[Dod03] P. E. Dodd and L. Massengill, “Dasic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics”, *IEEE TNS Nucl. Sci.*, vol. 50, no. 3, June 2003, pp. 583-600

[Dod04] P.E. Dodd, M.R. Shaneyfelt, J.A. Felix, and J.R. Schwank, “Production and Propagation of Single-Event Transients in High-Speed Digital Logic ICs”, *IEEE Trans. Nucl. Sci.*, vol. 51, Dec. 2004, pp. 3278 – 3284

[Dod96] P.E. Dodd, F.W. Sexton, G.L. Hash, M.R. Shaneyfelt, B.L. Draper, A.J. Farino, and R.S. Flores, “Impact of technology trends on SEU in CMOS SRAMs”, *IEEE Trans. Nucl. Sci.*, vol. 43, Dec. 1996, pp. 2797-2804

[Dre98] J. Dreibelbis, J. Barth, H. Kalter, and R. Kho, “Processor-based Built-In Self-Test for Embedded DRAM”, *IEEE Journal of Solid-State Circuits*, Nov. 1998, vol. 33, no. 11, pp. 1731-1740

[Dye04] C.S. Dyer, K. Hunter, S. Clucas, and A. Campbell, “Observation of the Solar Particle Events of October and November 2003 from CREDO and MPTB”, *IEEE Trans. Nucl. Sci.*, Dec. 2004, vol. 51, no. 6, pp. 3388-3393

[Eat04] P. Eaton, J. Benedetto, D. Mavis, K. Avery, M. Sibley, M. Gadlage, and T. Turflinger, “Single Event Transient Pulsewidth Temporal Latch Technique”, *IEEE Trans. Nucl. Sci.* Vol. 51, Dec. 2004, pp. 3365-3368

[Edm91] L.D. Edmonds, “A simple estimate of funneling-assisted charge collection”, *IEEE Trans. Nucl. Sci.*, Feb. 1991, vol. 37, pp. 828-833

[Fra07] F.J. Franco and R. Velazco, “A Portable Low-Cost SEU Evaluation Board for SRAMs”, *in proc. Spanish Conference on Electron Devices*, Jan. 31 – Feb. 2, 2007, pp. 165-168

[Ful99] E. Fuller, M. Caffrey, P. Blain, C. Carmichael, N. Khalsa, A. Salazar, “Radiation Test Results of the Virtex FPGA and ZBT SRAM for Space Based Reconfigurable Computing” *in proc. MAPLD 1999*, C_2, September 1999

[Gas06] G. Gasiot, D. Giot, and P. Roche “Alpha-Induced Multiple Cell Upsets in Standard and Radiation Hardened SRAMs Manufactured in a 65nm CMOS Technology”, *in proc. NSREC 2006*

[Gui09] G. Hubert, R. Velazco, P. Peronnard, “A Generic Platform for Remote Accelerated Tests and High Altitude SEU Experiments on Advanced ICs: Correlation with MUSCA SEP3 Calculation”, *in proc. 15th IEEE International On-Line Testing Symposium*, Sesimbra-Lisboa, Portugal, 24-26 June 2009, p. 180

[Gus06] M.S. Gussenhover, E.G. Mullen, and D.H. Brautigam, “Improved Understanding of the Earth’s Radiation Belts from the CRRES Satellite”, *IEEE Trans. Nucl. Sci.*, vol. 43, np. 2, April 1996

[Har90] R. Harboe-Sørensen, E.J. Daly, C.I. Underwood, J.Ward, and L. Adams, “The Behavior of Measured SEU at Low Altitude During Periods of high Solar Activity”, *IEEE Trans. Nucl. Sci.*, Dec. 90m vol. 37, no. 6, pp. 1938-1943

[Hei07] T. Heijmen, P. Roche, G. Gasiot, K.R. Rorbes, and D. Giot, “A Comprehensive Study on the Soft-Error Rate of Flip-Flops From 90-nm Production Libraries”, *IEEE Trans. on Device and Material Reliability*, March 2007, vol. 7, no. 1, pp. 84-96

[Hei09] T. Heijmen and J. Verwijst, “Altitude and Underground Real-Time SET Tests of Embedded SRAM”, presented at RADECS 2009, Buges, Belgium

[Het99] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajshi, “Logic BIST for large Industrial Designs: Real Issues and Case Studies”, *in proc. IEEE International Test Conference*, 1999, pp. 358-367

[Hsi81] C.M. Hsieh, P.C. Murley, and R.R. O’Brien, “Dynamics of charge collection from alpha-particles tracks in integrated circuits”, *in proc. IEEE Int. Reliability Phys. Symp.* 1981, pp. 38-42

[Hua99] C. T. Huang, J.R. Huang, C.F. Wu, C.W. Wu, and T.Y. Chang, “A programmable BIST core for embedded DRAM”, *IEEE Design and Test for Computer*, 1999, vol. 16, no. 1, pp. 59-70

[Hui04] L.M. Huisman, M. Kassab, and L. Pastel, “Data Mining Intefrated Circuit Fails with Fail Commonalities”, *in proc. International Test Conference*, Oct. 2004, pp. 203-212

[IEE05] IEEE 1500 Standard for Embedded Core Test (SECT), 2005

[IEE94] IEEE 1149.1 Standard Test Access Port and Boundary-Scan Architecture (JTAG), 1994

[Iye02] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, “Efficien Wrapper/TAM co-optimization for large SOCs”, *in proc. IEEE Design, Automation and Test in Europe Conference and Exhibition*, 2002, pp. 491-498

[Jac93] M. Jacomet, “Layout Dependent Fault Analysis and Test Synthesis for CMOS circuits”, *IEEE TCAD* 1993, pp. 888-889

[Kim07] D. Kim, I. Pomeranz, M.E. Amyeen, and S. Venkataraman, “Testing for Systematic Defects Based on DFM Guidelines”, International Test Conference Proceedings, 2007

[Kra03] K. Kranitis, G. Xenoulis, A. Pascalis, D. Gizopoulos, and Y. Zorian, *“*Application and Analysis of RT-Level Software-Based Self-Testing for Embedded Processor Cores”, *in proc. IEEE International Test Conference*, 2003, pp. 431-440

[Kra05] N. Kranitis, G. Xenoulis, A. Paschalis, and D. Gizopoulos “Software-based self-testing of embedded processors”, *IEEE Transactions on Computers*, vol. 54, vo. 4, April 2005, pp. 461-475

[Kru04] B. Kruseman, A. Majhi, C. Hora, S. Eichenberger, and J. Meirlevede, “Systematic Defects in Deep Sub-Micron Technologies”, *in proc. International Test Conference*, Oct. 2004, pp. 290-299

[Lim00] F.G. de Lima, E.C. Cota, L. Carro, M. Lubaszewski, R. Reis, R. Velazco, and S. Rezgui, “Designing a Radiation Hardened 8051-like Micro-controller”, *in proc. of 13th Symposium on Integrated Circuits and System Design*, 2000, pp. 255-260

[Lim01] F. Lima, C. Carmichael, J. Fabula, R. Padovani, and R. Reis, "A Fault Injection Analysis of Virtex FPGA TMR Design Methodology," *in proc. of the Radiation Effects on Components and Systems Conference (RADECS2001)*, Grenoble, FRANCE, 2001

[Lim02] F. Lima, L. Carro, R. Velazco, and R. Reis, “Injecting Multiple Upsets in a SEU Tolerant 8051 Micro-Controller”, *in proc. of 8th IEEE International On-Line Testing Workshop*, 2002

[Mad04] R. Madge, B. Benware, R. Turakhia, R. Daasch, C. Schuermyer, and J. Ruffler, "In Search of the Optimum Test Set-Adaptive Test Methods for Maximum Defect Coverage and Lowest Test Cost", *in proc. International Test Conference*, Oct. 2004, pp. 230-212

[Mar99] E.J. Marinissen, Y. Zorian, R. Kapur, T. Taylor, and L. Whestel, "Towards a Standard for Embedded Core Test: An Example", *in proc. IEEE International Test Conference*, 1999, pp. 616-627

[Mcl82] F.B. McLean and T.R. Oldham, "Charge funnelling in n and p-type Si substrates", *IEEE Trans. Nucl. Sci.*, Dec. 1982, vol. 29, pp. 2018-2023

[Mey74] P. Meyer, R. Ramary, and W.R. Weber, "Cosmic rays – astronomy with energetic particles", *Physics Today*, vol. 27, no. 10, 23, 1974

[Mil25] R.A. Millikan, presentation before the National Academy of Science, November 9, 1925, Madison, Wisconsin.

[Muk03] S.S. Mukherjee, C. Weaver, J. Emer, S.K. Reinhardt, and T. Austin "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor", *in proc. 36th IEEE International Symposium on Microarchitecture* 2003

[Nic01] B. Nicoleascu, R. Velazco, and M. Sonza Reorda, "Effectiveness and Limitations of Various Software Techniques for Soft Error Detection: A Comparative Study", *in proc. of 7th International On-Line Testing Workshop*, 2001, pp. 172-177

[Nic03] B. Nicolescu, P. Peronnard, R. Velazco, and Y. Savaria, "Efficiency of Transient Bit-Flips Detection by Software Means: A Complete Study", *in proc. of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'03)*

[Nig04] P. Nigh and A. Gattiker, "Random and Systematic Defect Analysis Using IDDQ Signature Analysis for Understanding Fails and Guiding Test Decisions", *in proc. International Test Conference*, Oct. 2004, pp. 309-318

[Nor96] E. Normand, "Single Event Effects in Avionics", *IEEE Trans. Nucl. Sci.*, April 1996, vol. 43, no. 2, pp. 461-474

[Per08] P. Peronnard, R. Ecoffet, M. Pignol, D. Bellin, and R. Velazco, "Predicting the SEU Error Rate through Fault Injection for a Complex Microprocessor", *in proc. of IEEE International Symposium on Industrial Electronics*, June 30 – July 2, 2008, pp. 2288-2292

[Pet81] E. Petersen, "Soft errors due to protons in the radiation belt", *IEEE Trans. Nucl. Sci.*, Dec. 1981, vol. 28, pp. 3981-3986

[Pic05] "PicoBlaze 8-bit Embedded Microcontroller User Guide", *Xilinx User Guide UG129*, 2005

[San08] A. Sanyal, S.M. Alam, S. Kundu, "A Built-In Self-Test Scheme for Soft Error Rate Characterization", *in proc. IEEE International On-Line Testing Symposium*, 2008, pp. 65-70

[Sei04] N. Seifert and N. Nelson, "Timing Vulnerability Factors of Sequentials", *IEEE TNS Nucl. Sci.*, vol. 4, Sep. 2004, pp. 516-522

[Shi02] P. Shivakuma, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic", *in proc. IEEE International Dependable System and Networks (DNS 2002)*

[Sma05] D.F. Smart and M.A. Shea, "Galactic Cosmic Radiation and Solar Energetic Particles", Chapter 6 in *Handbook of Geophysics and the Space Enviroment*, edited by A.S. Jursa, Hanscom, AFB, MA, 1985, pp. 6-10

[Ste05] L. Sterpone and M. Violante, "A New Analytical Approach to Estimate the Effects of SEUs in TMR Architecture Implemented Through SRAM-based FPGA", *IEEE Transactions on Nuclear Science*, 2005, Vol. 52, No. 6, December 2005, pp. 2217 – 2223

[Ste06_1] L. Sterpone, M. Violante, "A new reliability-oriented place and route algorithm for SRAM-based FPGAs", *IEEE Transactions on Computers*, Vol. 55, No. 6, June 2006, pp. 732 – 744

[Ste06_2] L. Sterpone, M. Violante, S. Rezgui, "An Analysis based on Fault Injection of Hardening Techniques for SRAM-based FPGAs", *IEEE Transactions on Nuclear Science*, Vol. 53, Issue 4, August 2006, pp. 2054 – 2059

[Str02] C.E. Stroud, "A designer's Guide to Built_In Self_Test", *Kluwer Academic Publisher*, 2002

[Swi04] G. M. Swift, "Virtex-II Static SEU Characterization," *Xilinx Radiation Test Consortium*, Tech. Rep. 1, 2004

[Tha80] S. Thatte, J. Abraham, "Test Generation for Microprocessors", *IEEE Trans. On Computer*, vol. c-29, June 1980, pp. 429-441

[Tmr04] "TMRTool User Guide", *Xilinx User Guide UG156*, 2004

[Tre93] R. Treuer and V.K. Agarwal, "Built-In Self Diagnosis for Repairable Embedded RAMs", *IEEE Design and Test for Computers*, June 1993, vol. 10, no. 2, pp. 24-33

[Tsa01] C.H. Tsai and C.W. Wu, "Processor-Programmable Memory BIST for Bus-Connected Embedded Memories", *in proc. Design Automation Conference*, 2001, pp. 352-330

[Tsa84] C.H. Tsao, R. Silberberg, and J.R. Letaw, "Cosmic Ray Heavy Ions at and above 40,000 feet", *IEEE Trans. Nucl. Sci.*, Dec. 1984, vol. 31, no. 6, pp. 1183-1185

[Van98] A.J. Van de Goor, "Testing Semiconductor Memories: Theory and Practice", *ComTex Publishing*, Gouda, The Netherlands, 1998

[Var00] F. Vargas, A. Amory, and R. Velazco, "Estimating Circuit Fault-Tolerance by Means of Transient-Fault Injection in VHDL", *in proc. of 6th IEEE International On-Line Testing Workshop*, July 2000, pp. 67-72

[Vio07] M. Violante, L. Sterpone, A. Manuzzato, S. Gerardin, P. Rech, M. Bagatin, A. Paccagnella, C. Andreani, G. Gorini, A. Pietropaolo, G. Cardarilli, S. Pontarelli, and C. Frost, "A New Hardware/Software Platform and a New 1/E Neutron Source for Soft Error Studies: Testing FPGAs at the ISIS Facility", *IEEE Trans. Nucl. Sci.*, vol. 54, issue 4, part 2, Aug. 2007, pp. 1184-1189

[Wea87] H.T. Weaver, C.L. Axness, J.S. Fu, J.S. Binkley, and J. Mansfield, "RAM cell recovery mechanisms following high-enery ion strikes", *IEEE Electron. Device Lett.*, Jan. 1987, vol. 8, pp. 7-9

[Wea88] H.T. Weaver, *"Soft error stability of p-well versus n-well CMOS latches derived from 2D, transient simulations"*, in IEDM Tech. Dig., 1988, pp. 512-515

[Wro00] F. Wrobel, J.M. Palau, M.C. Calvet, O. Bersillon, and H. Duarte, *"Incidence of multi-particle events on soft error rates caused by n-Si nuclear reactions"*, IEEE Trans. Nucl. Sci., Dec. 2000, vol. 47, pp- 2580-2585

[Zhu05] X. Zhu, R. Baumann, C. Pilch, J. Zhou, J. Jones, and C. Cirba, *"Comparison of Product Failure Rate to Component Soft Error Rate in a Multicore Digital Processor"*, in Proc. 43rd Int. Reliability Physics Symp. (IRPS), IEEE EDS, San Jose, CA, 2005, pp. 204-214

[Zie04] J. Ziegler, Helmut Puchner, *"SER – History, Trends and Challenges a Guide for Designing with Memory ICs"*, Cypress Semiconductor (2004)

[Zie08] J. Ziegler, "SRIM – The Stopping and Range of Ions in Matter", available: *http://www.srim.org*

[Zie96] J. F. Ziegler, H. W. Curtis, H. P. Muhlfeld, C. J. Montrose, B. Chin, M. Nicewicz, C. A. Russell, W. Y. Wang, L. B. Freeman, P. Hosier, L. E. LaFave, J. L. Walsh, J. M. Orro, G. J. Unger, J. M. Ross, T. J. O'Gorman, B. Messina, T. D. Sullivan, A. J. Sykes,

H. Yourke, T. A. Enger, V. Tolat, T. S. Scott, A. H. Taber, R. J. Sussman, W. A. Klein, and C. W. Wahaus, “IBM Experiments in Soft Fails in Computer Electronics (1978-1994)”, *IBM J. Research and Development*, vol. 40, no. 1, pp. 3-18, Jan. 1996

[Zor02] Y. Zorian, “What is an Infrastructure IP?”, *in proc. IEEE Design & Test of Computers*, vol. 19, no. 3, May-June 2002, pp. 5-7

Acknowledgments

This work wouldn’t have been done and this manuscript wouldn’t have been written without the help of many people I would like to deeply acknowledge both from a scientific and a human side. Parents, brothers, relatives, friends, colleagues, students: everyone must be remembered and thanked. The help these people gave me is boundless and I know that just a couple of wrote lines won’t be enough to repay it. The main thing I have learned during these years of studies, experiments, tests, and hard work is that good research could be done only collaborating, sharing ideas and results so to grow together. I want to thank all the people that trusted in me and gave me the chance to discover the fascinating world of research and collaboration.

It all started during prof. **Alessandro Paccagnella** lessons on Digital Integrated Circuits. He was able to whet my curiosity and introduce me to the world of research. During these years he always found the right words to say to encourage and support me. Moreover, he showed me how a good teacher should be and gave me the chance to live lots of adventures forcing me to do my best to face them.

Paolo Bernardi and **Michelangelo Grosso** are to be thanked as they showed me how a fruitful research should work. They have really understood what the Ph.D. stands for, and renewed my idea of collaboration. It’s just thanks to them if I have enjoyed so much the SoC research topic, if I have done my best without being scared of the obstacles to climb over, and if I am so proud of the results we have obtained.

Luckily enough I also had wonderful colleagues or, better, friends to talk and discuss with. **Alessio**, **Andrea**, **Marco**, **Nicola**, thanks to you everything was easier, and funnier. You know!

There are many people that supported me during these years, that stayed close to me and made me feel loved and valued. I will try to find the right words to express my feelings and appreciations. I know it’s not much, but it’s the best I can do, my gift is my words, and the following ones are for you.

« I vostri figli non sono figli vostri.
Sono figli e figlie della sete che la vita ha di sé stessa.
Essi vengono attraverso di voi, ma non da voi,
E benché vivano con voi non vi appartengono.

Potete donare loro amore ma non i vostri pensieri:
Essi hanno i loro pensieri.
Potete offrire rifugio ai loro corpi ma non alle loro anime:
Esse abitano la casa del domani, che non vi sarà concesso visitare neppure in sogno.
Potete tentare di essere simili a loro, ma non farvi simili a voi:
La vita procede e non s'attarda sul passato.
Voi siete gli archi da cui i figli, come frecce vive, sono scoccate in avanti.
L'Arciere vede il bersaglio sul sentiero dell'infinito, e vi tende con forza affinché le sue frecce vadano rapide e lontane.
Affidatevi con gioia alla mano dell'Arciere;
Poiché come ama il volo della freccia così ama la fermezza dell'arco.»

Khalil Gibran, “Il Profeta”

Ai miei **genitori**, per avermi sempre donato il loro amore ma non avermi mai imposto i loro pensieri, per avermi offerto un rifugio e non una prigione, e per aver cercato di indirizzare la mia freccia sul sentiero dell’infinito.

Ai miei **fratelli**, perché anche se ci scontriamo, se stiamo crescendo e ognuno prenderà la propria strada, se sono più le volte che ci intralciamo invece di aiutarci, alla fine sappiamo che ci ritroveremo, ci aiuteremo invece di intralciarci e le nostre strade si incroceranno sempre.

Alla zia **Germana**, per avermi insegnato a dare tutto il possibile per fare felici gli altri, per non essersi mai preoccupata di rinunciare ai propri desideri per vedere noi contenti. A **Irmà Flavia**, per avermi insegnato l’umiltà, la vera gioia di vivere, per avermi fatto capire che non bisogna mai avere paura del futuro, basta fidarsi, buttarsi. A mia santola **Giulietta** per farmi sentire privilegiato e coccolato, a mio santolo **Giovanni** per i consigli sul come criticare e affrontare il mondo che mi circonda. A tutti i cugini, zii e i parenti che formano la mia famiglia, ognuno ha avuto un ruolo importante nel farmi diventare quello che sono. Merito o colpa vostra, quindi.

«Sempre caro mi fu quest'ermo colle,
e questa siepe, che da tanta parte
dell'ultimo orizzonte il guardo esclude.
Ma sedendo e mirando, interminati
spazi di là da quella, e sovrumani
silenzi, e profondissima quïete
io nel pensier mi fingo, ove per poco
il cor non si spaura. E come il vento
odo stormir tra queste piante, io quello
infinito silenzio a questa voce
vo comparando: e mi sovvien l'eterno,
e le morte stagioni, e la presente
e viva, e il suon di lei. Così tra questa
immensità s'annega il pensier mio:
e il naufragar m'è dolce in questo mare»

Giacomo Leopardi, "L'infinito"

Al dirigente scolastico dell'istituto Cavanis, prof. **Alessandro Gatto**, per la fiducia che mi ha dato, per avermi dimostrato con i fatti il significato di passione nel proprio lavoro, perché riesce a mettere gli studenti sempre e comunque al primo posto, indipendentemente da quanto difficile possa essere.

A **Davide**, perché non perde mai il suo spirito positivista e la sua istintività, e perché mi ha insegnato che davanti agli studenti bisogna semplicemente essere se stessi, ad **Andrea Badalin**, perché non ci potevano essere tutor più illustri a cui addossare la mia inesperienza, per l'energia e la convinzione che mette in tutto quello che fa, a **Claudia Ceccato¸** per aver condiviso il primo anno di insegnamento con tutti i problemi, le incertezze e le soddisfazioni che porta con sé, a **Damiano Carlesso**, per come riesce a non far prevalere la sua presenza ma il suo operato, per avermi insegnato che davanti a certe cornici non ha senso sprecare troppe parole, a **Giancarlo Cunial** per come riesce a rinnovare il valore del ruolo di insegnante e di persona di cultura, ad **Alberto Bevilacqua**, a **Paolo Carrer**, ad **Angelo Vido**, a **Michela Richiedei**, e a tutti i colleghi che popolano la sala insegnanti, la segreteria e l'amministrazione del Cavanis, per aver reso questa avventura indimenticabile, per come riescono a creare e mantenere un clima di collaborazione in cui ognuno fa la sua parte e quello che ritiene più giusto per il futuro degli studenti.

A **Balbo**, **Bastasin**, **Battistin**, **Bonora**, **Bortignon**, **Caberlotto**, **Cappozzo**, **Dalla Costa**, **Dalla Santa**, **De Luca**, **Drago**, **Falda**, **Favero**, **Ferronato**, **Fregona**, **Gardin**, **Gasparetto**, **Grigolo**, **Guerini**, **Marcolin**, **Martini**, **Mascotto**, **Minuzzo**, **Pellegrinelli**, **Rizzo**, **Rizzotto,** **Tessarollo**, **Trento**, **Vendramini**, **Zangaro**, **Avogadro**, **Baruchello**, **Bassani**, **Basso**, **Berton**, **Biasion**, **Bonnier**, **Bordin**, **Botter**, **Bravo**, **Canil**, **Farina**, **Fietta**, **Fuga**, **Lovison**, **Memola**, **Morassuti**, **Pegoraro**, **Pellizzer**, **Pezzino**, **Piovesan**, **Rech**, **Rossato**, **Saran**, **Turra**, **Zanin** e a tutte le persone a cui ho avuto l'onore di fare da prof.. E' stata un'esperienza indimenticabile, che mi ha insegnato tanto e mi ha fatto crescere ancora di più. Mi avete fatto capire veramente il significato di responsabilità, di maturità e di comprensione. Essere dall'altra parte della cattedra non è facile, e vi ringrazio per tutte le soddisfazioni che mi avete dato, per tutte le volte che avete cercato di capirmi, per aver rinnovato la mia fiducia nei giovani e nel futuro. Vi ringrazio anche per le sfide, per tutte le volte che mi avete messo alla prova e mi avete fatto arrabbiare, che avete reso le cose difficili, perché mi avete spinto a dare il meglio di me.

«Principio qui potest esse vita "vitalis", ut ait Ennius, quae non in amici mutua benevolentia conquiescat? Quid dulcius quam habere quicum omnia audeas sic loqui ut tecum? Qui esset tantus fructus in prosperis rebus, nisi haberes qui illis aeque ac tu ipse gauderet? Adversas vero ferre difficile esset sine eo qui illas gravius etiam quam tu ferret. Amicitia res plurimas continet: quoquo te verteris, praesto est, nullo loco excluditur, numquam molesta est; itaque non aqua, non igni, ut aiunt, locis pluribus utimur quam amicitia. Quocirca et absentes adsunt et egentes abundant et imbecilli valent et, quod difficilius dictu est, mortui vivunt: tantum eos honos, memoria, desiderium prosequitur amicorum.»

In primo luogo come potrebbe essere una vita degna di essere vissuta che non si appaghi del reciproco affetto di un amico? Cosa c'è di più dolce che avere vicino qualcuno con cui tu possa parlare di qualunque cosa così come faresti con te stesso? E quale vantaggio ci sarebbe nella prosperità, se non avessi qualcuno che ne godesse con te? Certamente sarebbe difficile sopportare le avversità senza uno che le sopportasse con maggior forza di te. L'amicizia racchiude innumerevoli aspetti: dovunque tu sia diretto, essa è a tua disposizione, non è allontanata da nessun posto, non è mai inopportuna; perciò, non dell'acqua, non del fuoco, come dicono, ci serviamo in parecchie occasioni, quanto dell'amicizia. Per questo motivo anche chi non è presente c'è, e chi si trova in povertà, ha una ricchezza, e i malati sono in salute e, cosa che è piuttosto difficile a dirsi, i morti vivono: tanto li accompagna l'onore, il ricordo e il rimpianto degli amici.

Cicerone, "De Amicitia"

Ad **Antonella**, perché la sua vicinanza mi ha sempre dato sicurezza, perché ogni volta che penso al futuro lei, in qualche modo, c'è sempre.

Alla mia "sorellina" **Erica**, per come ci diamo forza a vicenda, per come riusciamo a sfogarci ridendo di quello che ci succede e degli ostacoli che dobbiamo e vogliamo superare, per come mi fa vedere le difficoltà sotto un'altra luce, rendendole affrontabili.

A **Filippo**, perché anche se mi ha fatto venire la "claudite", causa principe delle mie paranoie, mi ha aiutato molto quando ne avevo proprio bisogno. Spero di riuscire a ricambiare il favore, anche se ultimamente non mi sta riuscendo molto bene.

A **Paolo**, per il modo con cui affronta la vita e mette tutto se stesso in ogni piccola emozione ed esperienza che vive, per lo stile che ci ha sempre legato, per la sua naturalezza, e la sicurezza che mi da di avere un amico su cui contare, sempre.

A **Nicola**, per come è riuscito a farmi sentire importante, perché spero di avergli fatto capire quanto lui sia importante e che non c'è niente che non possa fare. Il tuo futuro devi costruirlo con le tue scelte, senza paura, sapendo che qualunque decisione prenderai, un giorno ti sembrerà la migliore. Goditi la vita.

A **Silvia**, perché mi ha insegnato che si deve continuare sempre a guardare avanti, a seguire le proprie convinzioni, a vivere pienamente, senza riserve, che non bisogna avere paura di cambiare e di fare anche un paio di passi indietro per riprendere il cammino verso l'alto.

A **Mav**, per le follie che solo noi possiamo fare, perché non si possono dimenticare le fatiche, i divertimenti e le soddisfazioni che abbiamo condiviso, a **Davide Z**, sperando abbia finalmente trovato la sua strada e riesca a percorrerla fino in fondo, per quel giro in moto che mi ha fatto capire quanto mi fido di lui, a **Bebo**, per come ci siamo sostenuti nelle difficoltà, per avermi costretto a crescere e per come sa capirmi. Il futuro è vostro ragazzi.

A **Marco C.**, per l'amicizia che mi ha sempre dimostrato, per essersi sempre fidato di me, a **Silvia** e **Irene**, le mie preferite, per come sanno farmi divertire con la loro ironia, la loro intelligenza e il legame che le unisce e che invidio un sacco.

Alla "zia" **Laura**, a **Beppe**, a **Matteo**, a **Massimo**, a **Nicola**, a **Fabiana**, ad **Alessandra Calore**, perché hanno reso l'avventura all'università di Padova degna di essere vissuta, per le difficoltà che abbiamo sopportato e le soddisfazioni che mi hanno aiutato a raggiungere e, soprattutto, per il sostegno che non mi hanno mai negato.

A **Beppe e Betta**, **Francesco ed Elisa**, **Domenico e Susanna**, perché guardandoli capisco cosa manca nella mia vita, ma che un giorno, sono sicuro, riuscirò a trovare anch'io, per l'invidia e l'ammirazione incommensurabile che provo nei loro confronti.

A **Mariangela**, **Itala**, **Piera**, **Antonella**, **Oriella**, **P. Graziano**, **Giovanni**, **Toni**, per essere degli ottimi esempi da seguire e delle gran persone che mi hanno insegnato che la vita può essere trasformata in qualcosa di ammirevole, di ironico, di spettacolare.

A **Teresa**, a **Marcela**, a **Manuelo**, per avermi fatto capire che non bisogna aver paura di fare delle scelte coraggiose, che la vita è donarsi, è amare, è fidarsi. Una parte di me è sempre con voi, e non tornerà mai più indietro.

A **Erica Z**, per come mi ha aiutato a migliorarmi, a capire che si può sbagliare anche quando si è convinti di aver fatto la cosa giusta, a **Tania**, perché ce l'abbiamo proprio fatta, anche se sembrava impossibile, al mio "fratellino" **Luca**, per come invidio il suo modo di vivere, di divertirsi, di affrontare le difficoltà, di essere amico, a **Francesco**, perché nei casini sa tirare fuori il meglio di me, sapendo che quando siamo io e lui nulla è impossibile, per come riesce ad essermi vicino anche quando siamo distanti, perché ogni volta che vedrò una stella cadente gli dedicherò un pensiero, a **Betta**, per i nostri discorsi filosofeggianti e il modo in cui riesce a trovare la bellezza in tutto quello che la circonda, a **Rossella**, per la passione per la musica, per la vita, quella vera, quella da vivere sorridendo, ballando, scatenandosi, a **Chiara**, a **Michela**, ad **Anna Botte**, a **Giulia Be**, a **Giulia Bo**, a **Silvia Bo**, per essere delle ragazze spettacolari, con stile, con classe, come non ce ne sono tante in giro, a **Nicolò**, a **Manuel**, a **Edoardo**, a **Damiano B.**, ad **Alberto**, per essere delle persone stupende, che sanno cavarsela in ogni situazione, per avermi sorpreso, fatto divertire, preoccupare. Grazie, perché mi rendete veramente orgoglioso del paese in cui sono vissuto finora.

A **Gloria** e **Anna**, per come riusciamo a ritrovarci anche dopo tanto tempo e colmare gli spazi fra noi in pochi minuti, perché affrontano la vita a testa alta e cercano di farmi capire come si fa a diventare grandi.

A **Beatrice**, per non rinnegare mai quello in cui crede, per combattere con tutte le sue forze senza perdersi d'animo, perché è grazie a persone così che il futuro sarà migliore.

A **DD**, per le eterne ma avvincenti radiocronache, per le frecciate mirate ed efficaci, per essere riuscito a rubarmi qualunque cosa da davanti agli occhi, a **Paolo**, per come riesce a vivere la vita fino in fondo, sperando riesca anche a rendere felici le persone che ci tengono a lui, a **Tex**, per avermi fatto capire che non ci si deve fermare alle apparenze, e per essere una persona seria, di cui mi fido senza ombra di dubbio, a **Lara**, per come sa preoccuparsi di ogni minimo dettaglio e auto criticarsi, sperando riesca a vivere serenamente con la consapevolezza di non avere nulla da invidiare a nessuno, ad **Ilaria**, per l'amore per la cultura e la letteratura che ha fatto rinascere in me, a **Mariaelena**, per come riesce a mettermi a disagio con la sua padronanza dell'italiano, per come mi faceva sentire quando alzava la mano per chiedere spiegazioni e quando la abbassa sulla tastiera del pianoforte, a **Davide**, sperando riesca a scoprire i propri sogni e a vivere senza rimpianti, a **Poldo**, sperando riesca a capire che per essere rispettati non serve fare i prepotenti, soprattutto quando si è una gran persona, a **Nick Martini**, per come custodisce gelosamente segreti che mi riguardano, ad **Alberto**, attore protagonista che sa sempre cavarsela e sempre con molto stile, a **Matteo**, perché possa decidere finalmente il suo futuro, a **Giorgio**, per avermi fatto capire prima di tutti gli altri quanto difficile sia essere un professore, e quanto sia impossibile far comprendere agli studenti che tutto quello che facciamo è per il loro bene, a **Zanga**, per essere pieno di energia, anche troppo a volte, ad **Augusto** per essere la follia fatta persona, a **Federico**, **Francesco Antonio**, **Enrico**, **Andrea**, **Guglielmo**, **Stefania**, **Giorgia**, **Silvia**, **Giulia**, **Chiara**, per avermi sopportato e per essersi fatti apprezzare come persone prima che come studenti, a **Nik**, per come sa farmi divertire e irritare allo stesso tempo, perché anche se non lo vuole ammettere la pensiamo allo stesso modo, e spero riuscirà a capire chi è veramente e a non aver paura di essere se stesso, 15315450197802813617810 15!! A **Deo**, per lasciarsi leggere nel pensiero e per come riesce a rendere divertente ogni momento, per avermi spiegato e dimostrato che un amico non può mai creare casini. Attento alla testa!

Ad **Aco**, per avermi insegnato tante cose, per avermi seguito in tante avventure e per cercare sempre di non farmi stare male, di non farmi sentire meno importante, di rende l'amicizia semplice, come dovrebbe essere, e per avermi dimostrato che effettivamente quello che conta non è quello che si dice, ma quello che si fa capire.

A **Carlo**, per i cineristori, le discussioni e le sfide che non manchiamo mai di affrontare, per essere un punto di riferimento indiscutibile e inestimabile, un protagonista e un amico che risolve tutto, anche se a modo suo, ma, soprattutto, per essere sempre a disposizione al momento giusto, a parte quando bisogna leggere.

A **Damiano**, perché anche se non riesco a trovare le parole per dirti quanto sei fondamentale so che lo capisci lo stesso, perché con te posso parlare di qualunque cosa così come ne parlerei a me stesso, posso condividere le soddisfazioni sapendo che ne sei contento tanto quanto me, i casini e le difficoltà sapendo che sai affrontarle con più forza di me e perché spero che la nostra amicizia sarà sempre la stessa, ovunque noi due saremo diretti.

Grazie, perché avete reso la mia vita degna di essere vissuta.