International Journal of Advances in Computer Science & Its Applications– IJCSIA
Volume 5: Issue 1 [ISSN : 2250-3765]
Publication Date : 30 April, 2015

Radiation Hardening: Approach and Reliability Impacts

Hycham Aboutaleb[1]* Bruno Monsuez[1]

***Abstract*— In aerospace field, it is necessary to address the radiation effects to which they are especially sensitive. Besides, the reuse of existing components implies that a modification of the architecture is sometimes expected to meet the safety requirements for such critical applications. To achieve such safety level, physical radiation hardening is usually used despite its cost. However, by performing a radiation hardening, it is possible to achieve the expected results while decreasing the expected cost of such an evolution. As a first step, a state of the art of existing mechanisms for logical radiation hardening is performed. These mechanisms are evaluated according a set of parameters: the type of errors they address, whether it is for purpose of detection or correction, the performance, the necessary additional physical volume, the computing time. To select the mechanisms to be used, a trade-off is performed, which depends also on the reliability analysis performed as well as the components that are concerned and on which the mechanisms are to be applied. A comparison between the unprotected module and the protected module is performed. The results obtained show that the optimized selection of hardening mechanisms yields to an improvement in reliability.**

***Keywords*—reliability, hardening techniques, trade-off, safety.**

I. Introduction

With the growing importance of embedded systems in aerospace field, it is necessary to address the radiation effects to which they are especially sensitive. Particularly, the embedded aerospace systems must take into account the disruptive and sometimes destructive effects of heavy ions.

Once radiation occurs, the doses received are different depending on whether one considers the spatial or atmospheric environment. Each type of particle (heavy ions, photons, light particles) has a different impact on components, as singular effect or cumulative effect. Depending on the type of component technology (bipolar, Mos ...) the effects of radiation are more or less destructive and fast. Singular effects can be non-destructive (SEU, SET) or destructive (Latch-Up, Burnout, rupture).

One of the best protections against radiation lies in the use of technology to limit the effects of radiation (RAD-Hard). There are three different and complementary approaches.

Hycham Aboutaleb[1]*, Bruno Monsuez[1]
[1]Computer Science and System Engineering Department, Ensta ParisTech, France

- Protection shield housing or chip can limit the effect of alpha particles.
- The radiation hardened circuits are often realized on an insulating material rather than on a semiconductor substrate (SOS or SOI for example). The sensitivity to the effects of dose and SEU is greatly diminished by this kind of technology.
- It is also possible to enhance protection while working on clean architecture transistors and elementary gates Technology (libraries)

The high cost of access to such technology Rad Hard reduces its use, which is limited today mainly for space applications.[1]

In addition to these purely technological approaches there exist architectural passivation techniques that limit the effect of radiation on embedded systems. The system redundancy techniques (e.g. triplication), protection of data stored in memory (detection / correction) are applicable to the system level, detect or correct some errors due to radiation. The scrubbing technique that consists in periodically reloading the configuration of an FPGA can be used in specific cases.

Finally, it is possible to apply architectural passivation techniques on the electronic components of FPGA or ASIC. The techniques described in the HDL code, have been designed to detect and possibly correct non-destructive effects (SEU) induced by the radiation received by the component [2]. Hardware redundancy techniques (duplication, triplication) applied in a targeted way can effectively protect the most critical parts of a component. Temporal redundancy techniques limit the extra cost surface, but penalize the protected function performance. Data protection techniques may be applied to internal memory spaces.

Adapted techniques also protect the heart of the components (the state machines) with the objective of detecting, reporting (monitoring) and sometimes correcting abnormal sequencing. Other architectural techniques can protect counters and registers of critical components. It is also possible to partially protect the combinatorial elements, decoders.

In this paper we intend to present techniques and technologies that limit the effect of most of the radiation. These approaches are seamlessly combined to increase the overall tightening of the application.

The second part introduces the main types of components that are studied as well as the main techniques that may be used to protect them. Then a third part is dedicated to the analysis of the main effects of applying such techniques. Finally a fourth part presents the results obtained when applying the selected techniques in an aerospace project from a reliability point of view

II. State of the art

A. Target Components

The selection of a protection technique depends on certain parameters, mainly those related to the system in which they operate. In order to be most effective in terms of level of protection and use of the matrix of FPGAs, each target category to protect will be divided into sub-categories according to their criticality. Different protection solutions can then be quickly implemented.

Targets to protect are those that seem the most critical to us and those that contain the most sensitive elements (ie switches, memory point ...). Four major targets families are identified:

- Memories: RAM, internal ROM.
- FSM: State Machines.
- Counters, sequential operators
- Registers.

B. Hardening Techniques

A lot of hardening techniques can be found in literature. [3][4][5] Each technique is first classified according to the error type it addresses and whether it only detects this error or detects and corrects it (see Table I). This parameter is important and shall be taken into account for reliability purposes.

TABLE I. LIST OF HARDENING TECHNIQUES AND CORRESPONDING ADDRESSED ERROR TYPES

List of techniques	Error type addressed	Detection	Correction
Hardware Duplication and comparison	SEU, SET	✔	
Hardware Duplication with redundancy	SEU, SET	✔	
Triple Modular Redundancy (TMR)	SEU, SET	✔	✔
Simple Temporal Redundancy	SEU	✔	
Repetition	SEU, SET	✔	✔
Double Coding	SEU, SET	✔	
Multiple Sampling	SET	✔	
Recomputing with shifted operands (RESO)	SEU	✔	
Recomputing with swapped operands (RESWO)	SEU	✔	
Recomputing duplication with comparison (REDWC)	SEU	✔	
Parity Bit	SEU	✔	
Hamming Code	SEU	✔	✔
Polynomial Codes	SEU	✔	
Convolutional Codes	SEU	✔	✔
Reed-Muller Codes	SEU	✔	✔
Turbo Codes	SEU	✔	✔
Low Density Parity Check	SEU	✔	✔
Cyclic Redundancy Check (CRC)	SEU	✔	
Reed-Solomon Codes	SEU	✔	✔
BCH Coding	SEU	✔	✔
Linear Digital State Variable	SEU	✔	✔

TABLE II. LIST OF HARDENING TECHNIQUES AND IMPACTS

List of techniques	Performance	Approximate additional surface	Approximate additional computing time
Hardware Duplication and comparison	Minimal	+100%	Minimal
Hardware Duplication with redundancy	Minimal	+100%	Minimal
Triple Modular Redundancy (TMR)	Average	+200%	Minimal
Simple Temporal Redundancy	Minimal	Low	+100%
Repetition	Average	Low	+200%
Double Coding	Average	Average	+100%
Multiple Sampling	Average	Average	Minimal
Recomputing with shifted oprands (RESO)	Average	Low	+100%
Recomputing with swapped operands (RESWO)	Average	Low	+100%
Recomputing duplication with comparaison (REDWC)	Average	Low	+100%
Parity Bit	Low	Low	Low
Hamming Code	Good	Average	Average
Polynomial Codes	Good	Average	Average
Onvolutifs Codes	Maximal	High	Average
Reed-Muller Codes	Good	Average	Average
Turbo Codes	Maximal	High	High
LowDensityParity Check	Maximal	Low	Low
Cyclic Redundancy Check (CRC)	Good	Low	Low
Reed-Solomon Codes	Good	Average	Average
BCH Coding	Good	Average	Average
Linear Digital State Variable	Average	Average	Low

There are three main categories of hardening techniques:

- Hardware techniques
- Software temporal techniques
- Software detecting/correcting codes

When applying a hardening technique, a number of modifications occur. First, the performance due to the introduction of the technique is impacted. Then, whether the technique is a hardware technique or a software technique software temporal technique or software detecting/correcting codes, it might be necessary to add surface. Moreover, hardening techniques imply an increase in computing time. It is expected and obvious that hardware techniques impact strongly the surface to add, while software techniques impact strongly the additional computing time. Each technique is therefore evaluated according to three main parameters (Table II):

- Performance
- Approximate additional surface
- Approximate additional computing time

III. Selection Methodology

Once all the hardening techniques are identified, the best protection techniques are selected according to a dependability study on the system to design in the aerospace project NADAE. The project NADAE intends to introduce CAN IP technology in aerospace industry. It is thus necessary to meet the safety requirements. These requirements can be found in certification recommendations such as DO 254, which is a guideline for airborne electronic hardware. As part of the project NADAE, an optimized selection of hardening techniques was performed. The best protection techniques are selected according to the CAN IP reliability study while keeping in mind the additional cost brought by these solutions in terms of occupancy of operating frequency in the component.

Seven major categories have been identified, depending on the component type as well as its criticality:

- FSM
  - *Selected technique*: Hamming code (8,4)
  - *Description*: The Hamming code provides very effective protection with a good performance for words with a size larger than 4-bit. Indeed, a 2-bit simultaneous error remains unlikely. In addition, the time overhead is low and the necessary additional surface too.
  - *Residual risk:* The Hamming code is effective only in case of a single error. If more than one error occurs, it is necessary to reset.
- Counters
  - *Selected technique*: Duplicated Gray code counter with a parity bit
  - *Description*: A SEU on a counter is detected by the parity bit and the second counter is selected.
  - *Residual risk:* This is optimal. It might still be ineffective if bit errors that have exactly the same position, which is unlikely.
- Small and highly critical registers
  - *Selected technique*: Hardware Triplication.
  - *Description*: The triplication allows a very efficient and fast protection for small registers.
  - *Residual risk:* The major setback is that the surface increases, leading to an increase of SEUs due to radiation.
- Large and highly critical registers
  - *Selected technique*: Hamming code (8,4).
  - *Description*: The Hamming code provides very effective protection with a good performance for words with a size larger than 4-bit. Indeed, a 2-bit simultaneous error remains unlikely. In addition, the time overhead is low and the necessary additional surface too.
  - *Residual risk:* The Hamming code is effective only in case of a single error. If more than one error occurs, it is necessary to reset.
- Moderately critical registers
  - *Selected technique*: Software Duplication with Parity bit.
  - *Description*: A SEU on a message is detected by the parity bit and the duplicate message is selected.
  - *Residual risk:* It might still be ineffective if bit errors that have exactly the same position, which is unlikely.
- Non critical registers
  - *Selected technique*: Parity bit
  - *Description*: It detects an error
  - *Residual risk:* If an error is detected, it cannot be corrected.
- RAM
  - The selected technique: Hamming code (8,4)
  - *Description*: The Hamming code provides very effective protection with a good performance for words with a size larger than 4-bit. Indeed, a 2-bit simultaneous error remains unlikely. In addition, the time overhead is low and the necessary additional surface too.
  - *Residual risk:* The Hamming code is effective only in case of a single error. If more than one error occurs, it is necessary to reset.

IV. Reliability Impacts

Once the selected techniques are applied to the architecture, reliability needs to be evaluated again.

In our project we have the following assumptions:

- The probability that the value of a bit is erroneously changed is the same for all registers, FSMs, counters and RAM.

- The passage of a bit from 0 to 1 and the passage of 1-0 due to an error have the same probability, ie P (0 → 1) = P (1 → 0).
- p is the probability a bit error occurs.
- N is the number of bits.

Since for a bit errors have the same probability, we have:

$$p = P(0 \rightarrow 1) + P(1 \rightarrow 0)$$

It is usually estimated that an error due to radiation has a probability of $p = 10^{-5}$

In the previous part, three main techniques were used: triplication, duplication with parity bit, Hamming code. We will evaluate the reliability for the component where these techniques are applied.

- Triplication:

An output is considered erroneous if two bits that have the same position are erroneous. Since the same bit is triplicated, the probability of an erroneous output bit is $3p^2$.

- Duplication with parity bit:

An output is considered erroneous if two bits that have the same position are erroneous. Since the same bit is duplicated, the probability of an erroneous output bit is p^2.

- Hamming Code:

An output is considered erroneous if at least two different bits are erroneous. The probability of an erroneous output is thus:

$$(1-((1-p)^N + Np(1-p)^{N-1}))/N$$

For $p = 10^{-5}$ and $N = 8$, we get: 3.5×10^{-10}

On higher level, after identifying the critical elements of a safety point of view, we apply the corresponding techniques as selected above, and subsequently assess criticality. To perform this analysis, we followed the following methodology:

1) Component Identification
2) Identification of functions associated with each component
3) Identification of undesired events for each function
4) Identification of the causes every dreaded event
5) Identification of the consequences for each undesired event
6) qualitative estimate of the severity of each undesired event
7) Estimation of the frequency of each undesired event
8) Deduction of the criticality of each undesired event
9) Extraction of the most dangerous undesired events
10) Deduction of severity of a SEU for each component
11) Deduction of the criticality of each component

The main undesired events are:

- Wrong Mode
- Message Not Received
- Reception of an erroneous message (standard mode)
- Reception of an erroneous message (extended mode)
- Message Not Transmitted
- Transmission of an erroneous message (standard mode)
- Transmission of an erroneous message (extended mode)
- Erroneous Synchronization

Before application of the protection techniques we have the following probabilities of occurrence for each undesired event:

Undesired Event	Probability before protection
Wrong mode	$2p$
Message Not Received	$5p+1-(1-p)^N$
Reception of an erroneous message (standard mode)	$80p$
Reception of an erroneous message (Extended Mode)	$104p$
Message Not Transmitted	$7p+1-(1-p)^N$
Transmission of an erroneous message (standard mode)	$80p$
Transmission of an erroneous message (Extended Mode)	$104p$
Erroneous synchronization	$20p$

After application of the protection techniques we have the following probabilities of occurrence for each undesired event:

Undesired Event	Probability after protection
Wrong mode	$6p^2$
Message Not Received	$15p^2+ (1 – ((1-p)^N + Np(1-p)^{N-1}))$
Reception of an erroneous message (standard mode)	$10(1 – ((1-p)^N + Np(1-p)^{N-1}))$
Reception of an erroneous message (Extended Mode)	$13(1 – ((1-p)^N + Np(1-p)^{N-1}))$
Message Not Transmitted	$21p^2+ (1 – ((1-p)^N + Np(1-p)^{N-1}))$
Transmission of an erroneous message (standard mode)	$10(1 – ((1-p)^N + Np(1-p)^{N-1}))$
Transmission of an erroneous message (Extended Mode)	$13(1 – ((1-p)^N + Np(1-p)^{N-1}))$
Erroneous synchronization	$60p^2$

By computing the probabilities with $p = 10^{-5}$ and $N = 8$, it is clear that the probability of an undesired event has decreased. The aim of the project was to increase the reliability by a factor 100. This was achieved at the component level.

v. Conclusion

With the growing importance of embedded systems in aerospace field, it is necessary to address the radiation effects to which they are especially sensitive. Particularly, the embedded aerospace systems must take into account the disruptive and sometimes destructive effects of heavy ions. [6]

Besides, the reuse of existing components implies that a modification of the architecture is sometimes expected to meet the safety requirements for such critical applications. To achieve such safety level, physical radiation hardening is usually used despite its cost. However, by performing a logical radiation hardening (by modifying optimally on the architecture), it is possible to achieve the expected results while decreasing the expected cost of such an evolution.

Once an existing module for the mentioned application is selected, it is necessary to enhance its reliability. As a first step, a state of the art of existing mechanisms for logical radiation hardening is performed. These mechanisms are evaluated according a set of parameters: the type of errors they address, whether it is for purpose of detection or correction, the performance, the necessary additional physical volume, the computing time. To select the mechanisms to be used, a trade-off is performed. This trade-off depends also on the safety and reliability analysis performed beforehand as well as the components that are concerned and on which the mechanisms are to be applied.

Once the most critical components are identified, logical radiation hardening is performed by selecting the optimal mechanism studied beforehand for each component. Trading-off at this level is necessary to meet the requirements while minimizing an expected increase in cost as well as in physical volume and in computing time. Seven types of components have been identified and the corresponding hardening mechanisms defined.

The residual risks are highlighted for each type of hardened components according to the hardening mechanism selected.

Once selected, the hardening mechanisms are integrated in the architecture and a new reliability analysis is performed to measure the impact of the "robustification" and verify that reliability has been indeed enhanced. A comparison between the unprotected module and the protected module is performed. The results obtained show that the optimized selection of hardening mechanisms yields to an improvement in reliability.

For future works, it would be interesting to compute the reliability on the higher level to evaluate the relevancy of the work presented in this paper on a higher scale. This could be done in parallel with test benches to verify if the expected results are met.

References

[1] R. Lacoc, "CMOS scaling, design principles and hardening-by-design methodologies", IEEE Nuclear and Space Radiation Effects Conference Short Course, Monterey, CA, July 2003.

[2] M.P. Baze, J.C. Killens R.A. Paup, and W.P. Snapp, "SEU Hardening Techniques for Retargetable, Scalable, Sub-Micron Digital Circuits and Librairies," Proceedings of 2002 SEE Symposium, Manhattan Beach, CA, April 2002.

[3] S. Habinc, "Suitability of reprogrammable FPGAs in space applications" Technical Report, Gaisler Research September 2002.

[4] G. Burke, S. Taft, "Fault Tolerant State Machines," Proceedings of MAPLD 2004, Washington D.C, USA, 2004.

[5] A. Jordan, "Evolution of a Fab-Independent Radiation-Hardened COTS IC Supplier", COTS Journal, November 2001.

[6] N. Perrot, M. Souyri, and J.F. Coldefy, "Circumventing Radiation Effects by Logic Design" Technical Report, Marconi Space Matra, July 1999.

About Author (s):

Hycham Aboutaleb is a PhD student in U2IS department at ENSTA-Paristech. He has a MSc in System Engineering from Ecole Polytechnique, France and a BSc in Electronics and Communications Engineering from Cairo University, Egypt. His current research areas include design methodologies, system engineering, safety engineering, complexity reduction, and quality control.

Bruno Monsuez is the Director of U2IS department at ENSTA-Paristech. He graduated from Ecole Polytechnique, France, where he obtained his PhD. His current research interests include developing compositional mathematical models used to represent hardware and software components of embedded systems and formal verification techniques for cojoint verification of the functional and nonfunctional properties.