Approximate hardening techniques for digital signal processing circuits against radiation-induced faults

by

Luis Ángel García Astudillo

A dissertation submitted by in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Electrical Engineering, Electronics and Automation

Universidad Carlos III de Madrid

Advisors:

Dr. Luis Entrena Arrontes
Dr. Almudena Lindoso Muñoz

Tutor:

Dr. Luis Entrena Arrontes

May 2023

This thesis is distributed under license “Creative Commons **Atributtion - Non Commercial - Non Derivatives**”.

The curse of climbing is discovering how great the distance yet to climb.

Steven Erikson

ACKNOWLEDGEMENTS

Me gustaría empezar esta Tesis con un agradecimiento a todas las personas que la han hecho posible, ayudándome a lo largo de mi etapa educativa e investigadora.

En primer lugar, quiero darles las gracias a mis tutores, el Dr. Luis Entrena y la Dra. Almudena Lindoso, por haberme dado la oportunidad de aprender de ellos. Su apoyo y su guía han sido fundamentales en todos los aspectos de esta Tesis y les estoy profundamente agradecido por haber confiado en mí. También me gustaría darle las gracias a la Dra. Marta Portela, que me introdujo en este apasionante campo durante mi Tesis de Master y me convenció para empezar esta Tesis Doctoral.

Quiero dar las gracias también al resto de profesores y personal administrativo del grupo de Diseño Microelectrónico y Aplicaciones por su apoyo, en especial al Dr. Mario García, cuyo apoyo técnico durante los experimentos ha sido inestimable.

A lo largo de estos años ha pasado mucha gente por el laboratorio en el que he trabajado. A todos ellos quiero agradecerles su compañía y buenos ratos y desearles buena suerte en sus carreras profesionales, a aquellos que ya se fueron, y en sus propias tesis doctorales, a aquellos que, estoy seguro, las acabarán pronto.

A mis padres, mi hermana y a Nerea les quiero agradecer su constante y desinteresado apoyo y cariño todos estos años, sin los cuales no habría llegado hasta aquí. Gracias.

Gracias también a mis amigos, que, aunque la adultez nos ha llevado a pasar menos tiempo juntos, han seguido sacando un rato para tomar algo, tirar unos dados y hablar de libros y música.

Por último, ahora que termino mi etapa como estudiante, que no mi etapa de aprendizaje, me gustaría acordarme de los profesores y profesoras de mis inicios, cuyas enseñanzas me marcaron para ser quien soy, en especial de Chelo, Rosalía, Fermín, Luis y Juan Carlos.

PUBLISHED AND SUBMITTED CONTENT

First-author published journal articles

[1] L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, *et al.*, “Analyzing Reduced Precision Triple Modular Redundancy under Proton Irradiation,” *IEEE Transactions on Nuclear Science*, vol. 69, pp. 470–477, 3 Mar. 2022. DOI: 10.1109/TNS.2022.3152088 (JCR Q2). This article has been wholly included in the Thesis in Chapter 3.

[2] L. Garcia-Astudillo, A. Lindoso, L. Entrena, *et al.*, “Error sensitivity study of FFT architectures implemented in FPGA,” *Microelectronics Reliability*, vol. 126, p. 114 298, Nov. 2021. DOI: 10.1016/J.MICROREL.2021.114298 (JCR Q3). This article has been wholly included in the Thesis in Chapter 4.

[3] L. A. Garcia-Astudillo, A. Lindoso, L. Entrena, *et al.*, “Analyzing Scaled Reduced Precision Redundancy for Error Mitigation under Proton Irradiation,” *IEEE Transactions on Nuclear Science*, pp. 1–1, 2022. DOI: 10.1109/TNS.2022.3147599 (JCR Q2). This article has been wholly included in the Thesis in Chapter 5.

[4] L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, *et al.*, “Reduced Resolution Redundancy: A Novel Approximate Error Mitigation Technique,” *IEEE Access*, vol. 10, pp. 20 643–20 651, 2022. DOI: 10.1109/ACCESS.2022.3152202 (JCR Q2). This article has been wholly included in the Thesis in Chapter 6.

[5] L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, *et al.*, “Evaluating Reduced Resolution Redundancy for radiation hardening,” *IEEE Transactions on Nuclear Science*, 2023, (Early Access). DOI: 10.1109/TNS.2023.3268825 (JCR Q2). This article has been accepted for publication and it is wholly included in the Thesis in Chapter 7.

Articles under review

[6] L. A. Garcia-Astudillo, A. Lindoso, and L. Entrena, “Error Mitigation using Optimized Redundancy for Composite Algorithms,” *IEEE Transactions on Aerospace and Electronic Systems*, 2023 (JCR Q1). The findings presented in this article have been summarized in Chapter 8 of this Thesis.

[7] P. Aviles, L. Garcia-Astudillo, J. Belloch, *et al.*, “Comparative of proton radiation data for 28 nm Zynq-7000 SoC,” *RADECS 2022 - European Conference on Radiation*

and its Effects on Components and Systems, 2022.

Conference articles

[8] L. A. Garcia-Astudillo, A. Lindoso, M. Portela, *et al.*, "Evaluation of a Reduced Precision Redundancy FFT Design," *2020 35th Conference on Design of Circuits and Integrated Systems, DCIS 2020*, Nov. 2020. DOI: 10.1109/DCIS51330.2020.9268634.

Whenever material from these sources is included in this Thesis, it is singled out with typographic means and an explicit reference.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Universidad Carlos III de Madrid's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

OTHER RESEARCH MERITS

During the course of this Thesis, the following research merits were awarded to the PhD candidate:

- RADSAGA and RADECS Association Student Grant to participate in the RADECS 2021 Conference.
- RADECS Association, RADSAGA and R2E Student Grant to participate in the RADECS 2022 Conference.

RESUMEN NO TÉCNICO

Se llama radiación al proceso por el cual una partícula o una onda es capaz de transmitir energía a través del espacio o un medio material. Si la energía transmitida es suficientemente alta, la radiación puede provocar que algunos electrones se desplacen de su posición, en un proceso llamado ionización.

La radiación ionizante puede provocar problemas a los seres vivos, pero también a los diversos materiales que componen los sistemas eléctricos y electrónicos utilizados en entornos sujetos a radiación. Existen en La Tierra varios procesos que emiten radiación ionizante, como la obtención de energía en centrales nucleares o ciertos procedimientos médicos. Sin embargo, las fuentes de radiación más importantes se sitúan más allá de nuestra atmósfera y afectan fundamentalmente a sistemas aeroespaciales y vuelos de gran altitud.

Debido a la radiación, los sistemas electrónicos que se exponen a cualquiera de estas fuentes sufren degradación en sus propiedades a lo largo del tiempo y pueden sufrir fallos catastróficos que acorten su vida útil. El envejecimiento de los componentes se produce por acumulación de carga eléctrica en el material, lo que se conoce como Dosis Ionizante Total (TID por sus siglas en inglés), o por distorsiones en el silicio sobre el que se fabrican los circuitos, lo que se conoce como Daño por Desplazamiento (DD). Una única partícula ionizante puede, sin embargo, provocar también diversos tipos de fallos transitorios o permanentes en los componentes de un circuito, generalmente por un cambio de estado en un elemento de memoria o fallos destructivos en un transistor. Los diferentes tipos de fallos producidos en circuitos por la acción de una única partícula ionizante se engloban en la categoría de Efectos de Evento Único (SEE por sus siglas en inglés).

Para proteger los sistemas electrónicos frente a los efectos de la radiación se suele recurrir a un conjunto de técnicas que llamamos endurecimiento frente a radiación. Los procedimientos tradicionales de endurecimiento han consistido en la fabricación de componentes electrónicos mediante procesos especiales que les confieran una resistencia inherente frente a la TID, el DD y los SEE. A este conjunto de técnicas de endurecimiento se lo conoce como Endurecimiento frente a la Radiación Por Proceso (RHBP por sus siglas en inglés). Estos procedimientos suelen aumentar el coste de los componentes y empeorar su rendimiento con respecto a los componentes que usamos en nuestros sistemas electrónicos cotidianos.

En oposición a las técnicas RHBP encontramos las técnicas de Endurecimiento frente a la Radiación Por Diseño (RHBD por sus siglas en inglés). Estas técnicas permiten detectar y tratar de corregir fallos producidos por la radiación introduciendo modificaciones en los circuitos. Estas modificaciones suelen aumentar la complejidad de los circuitos que se quiere endurecer, haciendo que consuman más energía, ocupen más espacio o funcionen a menor frecuencia, pero estas desventajas se pueden compensar con

la disminución de los costes de fabricación y la mejora en las prestaciones que aportan los sistemas modernos.

En un intento por reducir el coste de las misiones espaciales y mejorar sus capacidades, en los últimos años se trata de introducir un mayor número de Componentes Comerciales (COTS por sus siglas en inglés), endurecidos mediante técnicas RHBD.

Las técnicas RHBD habituales se basan en la adición de elementos redundantes idénticos al original, cuyos resultados se pueden comparar entre sí para obtener información acerca de la existencia de un error (si sólo se usa un circuito redundante, Duplicación Con Comparación [DWC]) o llegar incluso a corregir un error detectado de manera automática, si se emplean dos o más réplicas redundantes, siendo el caso más habitual la Redundancia Modular Triple (TMR) en todas sus variantes.

El trabajo desarrollado en esta Tesis gira en torno a las técnicas de endurecimiento RHBD de sistemas electrónicos comerciales. En concreto, se trata de proponer y caracterizar nuevas técnicas de endurecimiento que permitan reducir el alto consumo de recursos de las utilizadas habitualmente. Para ello, se han desarrollado técnicas de endurecimiento que aprovechan cálculos aproximados para detectar y corregir fallos en circuitos electrónicos digitales para procesamiento de señal implementados en FPGA comerciales, dispositivos que permiten implementar circuitos electrónicos digitales a medida y reconfigurarlos tantas veces como se quiera.

A lo largo de esta Tesis se han desarrollado diferentes circuitos de prueba endurecidos mediante TMR y se ha comparado su rendimiento con los de otras técnicas de Redundancia Aproximada, en concreto la Redundancia de Precisión Reducida (RPR), la Redundancia de Resolución Reducida (RRR) y la Redundancia Optimizada para Algoritmos Compuestos (ORCA):

- La Redundancia de Precisión Reducida se basa en la utilización de dos réplicas redundantes que calculan resultados con un menor número de bits que el circuito original. De este modo se pueden disminuir los recursos necesitados por el circuito, aunque las correcciones en caso de fallo son menos precisas que en el TMR. En este trabajo exploramos también la RPR Escalada como un método de obtener un balance óptimo entre la precisión y el consumo de recursos.

- La Redundancia de Resolución Reducida es una técnica propuesta originalmente en esta tesis. Está pensada para algoritmos que trabajan con información en forma de paquetes cuyos datos individuales guardan alguna relación entre sí. Las réplicas redundantes calculan los resultados con una fracción de los datos de entrada originales, lo que reduce su tamaño y permite correcciones aproximadas en caso de fallo.

- La Redundancia Optimizada para Algoritmos Compuestos es también una aportación original de esta tesis. Está indicada para algoritmos cuyo resultado final puede expresarse como la composición de resultados intermedios calculados en etapas anteriores. Las réplicas redundantes se forman como bloques que calculan resultados

intermedios y el resultado de su composición se puede comparar con el resultado original. Este método permite reducir recursos y proporciona resultados de corrección exactos en la mayor parte de los casos, lo que supone una mejora importante con respecto a las correcciones de los métodos anteriores.

La eficacia de las técnicas de endurecimiento desarrolladas se ha probado mediante experimentos de inyección de fallos y mediante ensayos en instalaciones de aceleradores de partículas preparadas para la irradiación de dispositivos electrónicos. En concreto, se han realizado ensayos de radiación con protones en el Centro Nacional de Aceleradores (CNA España), el Paul Scherrer Institut (PSI, Suiza) y ensayos de radiación con neutrones en el laboratorio ISIS Neutron and Muon Source (ChipIR, Reino Unido).

NON-TECHNICAL SUMMARY

Radiation is the process by which a particle or wave is able to transmit energy through space or a material medium. If the energy transmitted is high enough, radiation can cause some electrons to move out of position, in a process called ionization.

Ionizing radiation can cause problems for living beings, but also for the various materials that make up electrical and electronic systems used in environments subject to radiation. There are several processes on Earth that emit ionizing radiation, such as obtaining energy in nuclear power plants or certain medical procedures. However, the most important sources of radiation are located beyond our atmosphere and mainly affect aerospace systems and high-altitude flights.

Due to radiation, electronic systems exposed to any of these sources suffer degradation in their properties over time and can suffer catastrophic failures that shorten their useful life. Component aging is caused by accumulation of electrical charge in the material, known as Total Ionizing Dose (TID), or by distortions in the silicon on which the circuits are fabricated, known as Displacement Damage (DD). A single ionizing particle can, however, also cause various types of transient or permanent failures in the components of a circuit, usually by a change of state in a memory element or destructive failures in a transistor. The different types of failures produced in circuits by the action of a single ionizing particle fall into the category of Single Event Effects (SEE).

To protect electronic systems against the effects of radiation, a set of techniques known as radiation hardening is commonly used. Traditional hardening procedures have consisted of manufacturing electronic components using special processes that give them inherent resistance to TID, DD and SEE. This set of hardening techniques is known as Radiation Hardening By Process (RHBP). These procedures tend to increase the cost of components and worsen their performance relative to the components we use in our everyday electronic systems.

Opposed to RHBP techniques we find Radiation Hardening By Design (RHBD) techniques. These techniques allow us to detect faults and attempt to correct them by inserting modifications in the circuit. These modifications typically increase the complexity of the circuits to be hardened, causing them to consume more power, use more area or operate at lower frequencies, but these disadvantages can be compensated by the lower costs and improved performance provided by modern systems.

In an attempt to reduce the cost of space missions and improve their capabilities, efforts have been made in recent years to introduce more Commercial Off-The-Shelf (COTS) components, hardened with RHBD approaches. The usual RHBD techniques are based on the addition of redundant elements identical to the original, whose results can be compared with each other to obtain information about the existence of an error

(if only one redundant circuit is used, Duplication With Comparison [DWC]) or even correct an error detected automatically, if two or more redundant replicas are used, the most common case being Triple Modular Redundancy (TMR) in all its variants.

The work developed in this Thesis revolves around RHBD hardening techniques for commercial electronic systems. Specifically, the aim is to propose and characterize new hardening techniques to reduce the high resource consumption of those commonly used. For this purpose, we have developed hardening techniques that take advantage of approximate calculations to detect and correct faults in digital electronic circuits for signal processing implemented in commercial FPGA, devices that allow to implement custom digital electronic circuits and reconfigure them as many times as desired.

Throughout this Thesis, different TMR-hardened test circuits have been developed and their performance has been compared with those of other Approximate Redundancy techniques, namely Reduced Precision Redundancy (RPR), Reduced Resolution Redundancy (RRR) and Optimized Redundancy for Composite Algorithms (ORCA):

- Reduced Precision Redundancy is based on the use of two redundant replicas that compute results with a smaller number of bits than the original circuit. In this way, the resources needed by the circuit can be decreased, although the corrections in case of failure are less accurate than in TMR. In this paper we also explore Scaled RPR as a method of obtaining an optimal balance between accuracy and resource consumption.
- Reduced Resolution Redundancy is a technique originally proposed in this thesis. It is intended for algorithms that work with information in the form of packets whose individual data have some relationship with each other. Redundant replicas compute results with a fraction of the original input data, which reduces their size and allows approximate corrections in case of failure.
- Optimized Redundancy for Composite Algorithms is also an original contribution of this thesis. It is suitable for algorithms whose final result can be expressed as the composition of intermediate results computed in previous stages. Redundant replicas are formed as blocks that compute intermediate results and the result of their composition can be compared with the original result. This method reduces resources and provides accurate correction results in most cases, which is a significant improvement over the corrections of previous methods.

The effectiveness of the developed hardening techniques has been tested by fault injection experiments and by testing in particle accelerator facilities prepared for irradiation of electronic devices. In particular, proton irradiation campaigns have been carried out at Centro Nacional de Aceleradores (CNA, Spain) and at Paul Scherrer Institut (PSI, Switzerland), and neutron irradiation campaigns have been performed at the ISIS Neutron and Muon Source laboratory (ChipIR, United Kingdom).

RESUMEN TÉCNICO

Se llama radiación al proceso por el cual una partícula o una onda es capaz de transmitir energía a través del espacio o un medio material. Si la energía transmitida es suficientemente alta, la radiación puede provocar que algunos electrones se desplacen de su posición, en un proceso llamado ionización.

La radiación ionizante puede provocar problemas a los seres vivos, pero también a los diversos materiales que componen los sistemas eléctricos y electrónicos utilizados en entornos sujetos a radiación. Existen en La Tierra varios procesos que emiten radiación ionizante, como la obtención de energía en centrales nucleares o ciertos procedimientos médicos. Sin embargo, las fuentes de radiación más importantes se sitúan más allá de nuestra atmósfera y afectan fundamentalmente a sistemas aeroespaciales y vuelos de gran altitud.

Debido a la radiación, los sistemas electrónicos que se exponen a cualquiera de estas fuentes sufren degradación en sus propiedades a lo largo del tiempo y pueden sufrir fallos catastróficos que acorten su vida útil. El envejecimiento de los componentes se produce por acumulación de carga eléctrica en el material, lo que se conoce como Dosis Ionizante Total (TID, Total Ionizing Dose), o por distorsiones acumuladas en la matriz cristalina del silicio en el que se fabrican los circuitos, lo que se conoce como Daño por Desplazamiento (DD, Displacement Damage). Una única partícula ionizante puede, sin embargo, provocar también diversos tipos de fallos transitorios o permanentes en los componentes de un circuito, generalmente por un cambio de estado en un elemento de memoria o la activación de circuitos parasitarios en un transistor. Los diferentes tipos de fallos producidos en circuitos por la acción de una única partícula ionizante se engloban en la categoría de Efectos de Evento Único (SEE, Single Event Effects).

Para proteger los sistemas electrónicos frente a los efectos de la radiación se suele recurrir a un conjunto de técnicas que llamamos endurecimiento frente a radiación. Los procedimientos tradicionales de endurecimiento han consistido en la fabricación de componentes electrónicos mediante procesos especiales que les confieran una resistencia inherente frente a la TID, el DD y los SEE. A este conjunto de técnicas de endurecimiento se lo conoce como Endurecimiento frente a la Radiación Por Proceso (RHBP, por sus siglas en inglés). Estos procedimientos suelen aumentar el coste de los componentes y empeorar su rendimiento con respecto a los componentes que usamos en nuestros sistemas electrónicos cotidianos.

En oposición a las técnicas RHBP encontramos las técnicas de Endurecimiento frente a la Radiación Por Diseño (RHBD, por sus siglas en inglés). Estas técnicas permiten detectar y tratar de corregir fallos producidos por la radiación introduciendo modificaciones en los circuitos. Estas modificaciones suelen aumentar la complejidad de los circuitos que se quiere endurecer, haciendo que consuman más energía, ocupen más

espacio o funcionen a menor frecuencia, pero estas desventajas se pueden compensar con la disminución de los costes de fabricación y la mejora en las prestaciones que aportan los sistemas modernos.

En un intento por reducir el coste de las misiones espaciales y mejorar sus capacidades, en los últimos años se trata de introducir un mayor número de Componentes Comerciales (COTS, por sus siglas en inglés), endurecidos mediante técnicas RHBD.

Las técnicas RHBD habituales se basan en la adición de elementos redundantes idénticos al original, cuyos resultados se pueden comparar entre sí para obtener información acerca de la existencia de un error (si sólo se usa un circuito redundante, Duplicación Con Comparación [DWC, Duplication With Comparison]) o llegar incluso a corregir un error detectado de manera automática, si se emplean dos o más réplicas redundantes, siendo el caso más habitual la Redundancia Modular Triple (TMR, Triple Modular Redundancy) en todas sus variantes.

El trabajo desarrollado en esta Tesis gira en torno a las técnicas de endurecimiento RHBD de sistemas electrónicos comerciales. En concreto, se trata de proponer y caracterizar nuevas técnicas de endurecimiento que permitan reducir el alto consumo de recursos de las técnicas utilizadas habitualmente. Para ello, se han desarrollado técnicas de endurecimiento que aprovechan cálculos aproximados para detectar y corregir fallos en circuitos electrónicos digitales para procesamiento de señal implementados en FPGA (Field Programmable Gate Array) comerciales.

Las FPGA son dispositivos que permiten implementar circuitos electrónicos digitales diseñados a medida y reconfigurarlos tantas veces como se quiera. Su capacidad de reconfiguración y sus altas prestaciones las convierten en dispositivos muy interesantes para aplicaciones espaciales, donde realizar cambios en los diseños no suele ser posible una vez comenzada la misión. La reconfigurabilidad de las FPGA permite corregir en remoto posibles problemas en el diseño, pero también añadir o modificar funcionalidades a los circuitos implementados en el sistema.

La eficacia de las técnicas de endurecimiento desarrolladas e implementadas en FPGAs se ha probado mediante experimentos de inyección de fallos y mediante ensayos en instalaciones de aceleradores de partículas preparadas para la irradiación de dispositivos electrónicos.

Los ensayos de radiación son el estándar industrial para probar el comportamiento de todos los sistemas electrónicos que se envían a una misión espacial. Con estos ensayos se trata de emular de manera acelerada las condiciones de radiación a las que se verán sometidos los sistemas una vez hayan sido lanzados y determinar su resistencia a TID, DD y/o SEEs. Dependiendo del efecto que se quiera observar, las partículas elegidas para la radiación varían, pudiendo elegirse entre electrones, neutrones, protones, iones pesados, fotones... Particularmente, los ensayos de radiación realizados en este trabajo, tratándose de un estudio de técnicas de endurecimiento para sistemas electrónicos digitales, están destinados a establecer la sensibilidad de los circuitos estudiados frente a un tipo de SEE

conocido como Single Event Upset (SEU), en el que la radiación modifica el valor lógico de un elemento de memoria. Para ello, hemos recurrido a experimentos de radiación con protones en el Centro Nacional de Aceleradores (CNA, España), el Paul Scherrer Institut (PSI, Suiza) y experimentos de radiación con neutrones en el laboratorio ISIS Neutron and Muon Source (ChipIR, Reino Unido).

La sensibilidad de un circuito suele medirse en términos de su sección eficaz (*cross section*) con respecto a una partícula determinada, calculada como el cociente entre el número de fallos encontrados y el número de partículas ionizantes por unidad de área utilizadas en la campaña de radiación. Esta métrica sirve para estimar el número de fallos que provocará la radiación a lo largo de la vida útil del sistema, pero también para establecer comparaciones que permitan conocer la eficacia de los sistemas de endurecimiento implementados y ayudar a mejorarlos.

El método de inyección de fallos utilizado en esta Tesis como complemento a la radiación se basa en modificar el valor lógico de los datos almacenados en la memoria de configuración de la FPGA. En esta memoria se guarda la descripción del funcionamiento del circuito implementado en la FPGA, por lo que modificar sus valores equivale a modificar el circuito. En FPGAs que utilizan la tecnología SRAM en sus memorias de configuración, como las utilizadas en esta Tesis, este es el componente más sensible a la radiación, por lo que es posible comparar los resultados de la inyección de fallos y de las campañas de radiación. Análogamente a la sección eficaz, en experimentos de inyección de fallos podemos hablar de la tasa de error, calculada como el cociente entre el número de fallos encontrados y la cantidad de bits de memoria inyectados.

A lo largo de esta Tesis se han desarrollado diferentes circuitos endurecidos mediante Redundancia Modular Triple y se ha comparado su rendimiento con los de otras técnicas de Redundancia Aproximada, en concreto la Redundancia de Precisión Reducida (RPR), la Redundancia de Resolución Reducida (RRR) y la Redundancia Optimizada para Algoritmos Compuestos (ORCA). Estas dos últimas son contribuciones originales presentadas en esta Tesis.

- La Redundancia de Precisión Reducida se basa en la utilización de dos réplicas redundantes que calculan resultados con un menor número de bits que el circuito original. Para cada dato de salida se comparan el resultado del circuito original y los dos resultados de precisión reducida. Si los dos resultados de precisión reducida son idénticos y su diferencia con el resultado de precisión completa es mayor que un determinado valor umbral, se considera que existe un fallo en el circuito original y se utiliza el resultado de precisión reducida para corregirlo. En cualquier otro caso, el resultado original se considera correcto, aunque pueda contener errores tolerables por debajo del umbral de comparación. En comparación con un circuito endurecido con TMR, los diseños RPR utilizan menos recursos, debido a la reducción en la precisión de los cálculos de los circuitos redundantes. No obstante, esto también afecta a la calidad de los resultados obtenidos cuando se corrige un error.

En este trabajo exploramos también la RPR Escalada como un método de obtener un balance óptimo entre la precisión y el consumo de recursos. En esta variante de la técnica RPR, los resultados de cada etapa de cálculo en los circuitos redundantes tienen una precisión diferente, incrementándose hacia las últimas etapas, en las que el resultado tiene la misma precisión que el circuito original. Con este método se logra incrementar la calidad de los datos corregidos a la vez que se reducen los recursos utilizados por el endurecimiento.

Los resultados de las campañas de radiación y de inyección de fallos realizadas sobre los diseños endurecidos con RPR sugieren que la reducción de recursos no sólo es beneficiosa por sí misma en términos de recursos y energía utilizados por el sistema, sino que también conlleva una reducción de la sensibilidad de los circuitos, medida tanto en cross section como en tasa de error.

- La Redundancia de Resolución Reducida es una técnica propuesta originalmente en esta tesis. Está indicada para algoritmos que trabajan con información en forma de paquetes cuyos datos individuales guardan alguna relación entre sí, como puede ser un algoritmo de procesamiento de imágenes. En la técnica RRR, se añaden dos circuitos redundantes que calculan los resultados con una fracción de los datos de entrada originales. Tras el cálculo, los resultados diezmados pueden interpolarse para obtener un resultado aproximado del mismo tamaño que el resultado del circuito original. Una vez interpolados, los resultados de los tres circuitos pueden ser comparados para detectar y corregir fallos de una manera similar a la que se utiliza en la técnica RPR. Aprovechando las características del diseño hardware, la disminución de la cantidad de datos que procesan los circuitos de Resolución Reducida puede traducirse en una disminución de recursos, en lugar de una disminución de tiempo de cálculo. De esta manera, la técnica RRR es capaz de reducir el consumo de recursos en comparación a los que se necesitarían si se utilizase un endurecimiento TMR.

  Los resultados de los experimentos realizados en diseños endurecidos mediante Redundancia de Resolución Reducida sugieren que la técnica es eficaz en reducir los recursos utilizados y, al igual que pasaba en el caso de la Redundancia de Precisión Reducida, también su sensibilidad se ve reducida, comparada con la sensibilidad del mismo circuito endurecido con Redundancia Modular Triple. Además, se observa una reducción notable de la sensibilidad de los circuitos frente a errores no corregibles, comparado con el mismo resultado en TMR y RPR. Este tipo de error engloba aquellos producidos por fallos en la lógica de comparación y votación o aquellos en los que un único SEU produce fallos en los resultados de dos o más de los circuitos redundantes al mismo tiempo, lo que se conoce como Fallo en Modo Común (CMF). No obstante, también se observa que la calidad de las correcciones realizadas utilizando este método empeora ligeramente.

- La Redundancia Optimizada para Algoritmos Compuestos es también una aportación original de esta tesis. Está indicada para algoritmos cuyo resultado final puede

expresarse como la composición de resultados intermedios calculados en etapas anteriores. Para endurecer un circuito usando esta técnica, se añaden dos circuitos redundantes diferentes entre sí y que procesan cada uno una parte diferente del conjunto de datos de entrada. Cada uno de estos circuitos aproximados calcula un resultado intermedio. La composición de los dos resultados intermedios da un resultado idéntico al del circuito original en ausencia de fallos.

La detección de fallos se realiza comparando el resultado del circuito original con el de la composición de los circuitos aproximados. En caso de ser diferentes, se puede determinar el origen del fallo comparando los resultados aproximados intermedios frente a un umbral. Si la diferencia entre los resultados intermedios supera el umbral, significa que el fallo se ha producido en uno de los circuitos aproximados y que el resultado de la composición no debe ser utilizado en la salida. Al igual que ocurre en la Redundancia de Precisión Reducida y la Redundancia de Resolución Reducida, utilizar un umbral de comparación implica la existencia de errores tolerables. No obstante, esta técnica de endurecimiento permite realizar correcciones exactas, en lugar de aproximadas, en la mayor parte de los casos, lo que mejora la calidad de los resultados con respecto a otras técnicas de endurecimiento aproximadas, al tiempo que reduce los recursos utilizados por el sistema endurecido en comparación con las técnicas tradicionales.

Los resultados de los experimentos realizados con diseños endurecidos mediante Redundancia Optimizada para Algoritmos Compuestos confirman que esta técnica de endurecimiento es capaz de producir correcciones exactas en un alto porcentaje de los eventos. Su sensibilidad frente a todo tipo de errores y frente a errores no corregibles también se ve disminuida, comparada con la obtenida con Redundancia Modular Triple.

Los resultados presentados en esta Tesis respaldan la idea de que las técnicas de Redundancia Aproximada son alternativas viables a las técnicas de endurecimiento frente a la radiación habituales, siempre que las características del sistema toleren la presencia de resultados no exactos. La Redundancia Aproximada logra reducir el área y el consumo de recursos de los circuitos endurecidos a costa de realizar correcciones aproximadas, que van desde obtener menor precisión en los resultados corregidos (RPR), correcciones basadas en resultados correctos próximos (RRR) o correcciones exactas en algunos casos y de menor precisión en otros (ORCA). En cualquiera de los casos, estas correcciones aproximadas pueden ser percibidas como mero ruido por el sistema y no afectar a su funcionalidad.

Las campañas de radiación y de inyección de fallos muestran que las técnicas de Endurecimiento Aproximado reducen la sensibilidad de los circuitos endurecidos (con respecto a la del mismo circuito endurecido con Redundancia Modular Triple) como consecuencia de la reducción de recursos. Comparar los resultados de ambos tipos de campañas nos ha permitido establecer una relación entre ellas, lo que valida el método de inyección de fallos y lo sitúa como una manera rápida y económica de obtener

resultados preliminares. La inyección de fallos nos ha permitido, por un lado, depurar el funcionamiento de las técnicas propuestas sin necesidad de someterlas a ensayos de radiación o largas campañas de simulación, y por otro, conocer de antemano la sensibilidad y capacidad de corrección de errores de las técnicas de endurecimiento propuestas. Estas dos ventajas nos han permitido realizar una planificación de las campañas de radiación más adecuada, aprovechando al máximo el tiempo de radiación disponible y minimizando los costes de los experimentos.

Además de estas características, los resultados obtenidos sugieren que la manera en que las técnicas de Redundancia de Resolución Reducida y en la Redundancia Optimizada para Algoritmos Compuestos tratan los datos reduce la incidencia de Fallos en Modo Común, que son responsables de gran parte de los fallos no corregibles en todos los sistemas endurecidos mediante técnicas RHBD.

TECHNICAL SUMMARY

Radiation is the process by which a particle or wave is able to transmit energy through space or a material medium. If the energy transmitted is high enough, radiation can cause some electrons to move out of position, in a process called ionization.

Ionizing radiation can cause problems for living beings, but also for the various materials that make up electrical and electronic systems used in environments subject to radiation. There are several processes on Earth that emit ionizing radiation, such as obtaining energy in nuclear power plants or certain medical procedures. However, the most important sources of radiation are located beyond our atmosphere and mainly affect aerospace systems and high-altitude flights.

Due to radiation, electronic systems exposed to any of these sources suffer degradation in their properties over time and can suffer catastrophic failures that shorten their useful life. Component aging is caused by accumulation of electrical charge in the material, known as Total Ionizing Dose (TID), or by accumulated distortions in the crystalline matrix of the silicon in which the circuits are made, known as Displacement Damage (DD). A single ionizing particle can, however, also cause various types of transient or permanent failures in the components of a circuit, usually by a change of state in a memory element or the activation of parasitic circuits in a transistor. The different types of faults produced in circuits by the action of a single ionizing particle fall into the category of Single Event Effects (SEE).

To protect electronic systems against the effects of radiation, a set of techniques known as radiation hardening is often used. Traditional hardening procedures have consisted of manufacturing electronic components using special processes that give them inherent resistance to TID, DD and SEE. This set of hardening techniques is known as Radiation Hardening By Process (RHBP). These procedures tend to increase the cost of components and worsen their performance relative to the components we use in our everyday electronic systems.

In opposition to RHBP techniques we find Radiation Hardening by Design (RHBD) techniques. These techniques allow us to detect and attempt to correct radiation induced failures by introducing modifications to the circuits. These modifications often increase the complexity of the circuits to be hardened, causing them to consume more power, take up more space or operate at a lower frequency. However, these disadvantages can be offset by the reduced manufacturing costs and improved performance provided by modern systems.

In an attempt to reduce the cost of space missions and improve their capabilities, efforts have been made in recent years to introduce more Commercial Off-The-Shelf (COTS) components, hardened with RHBD approaches.

The usual RHBD techniques are based on the addition of redundant elements identical to the original, whose results can be compared with each other to obtain information about the existence of an error (if only one redundant circuit is used, Duplication With Comparison [DWC]) or even to correct an error detected automatically, if two or more redundant replicas are used, the most common case being Triple Modular Redundancy (TMR) in all its variants.

The work developed in this Thesis revolves around RHBD hardening techniques for commercial electronic systems. Specifically, the aim is to propose and characterize new hardening techniques to reduce the high resource consumption of commonly used techniques. To this end, we have developed hardening techniques that take advantage of approximate calculations to detect and correct faults in digital electronic circuits for signal processing implemented in commercial FPGAs (Field Programmable Gate Array).

FPGAs are devices that allow custom-designed digital electronic circuits to be implemented and reconfigured as many times as desired. Their reconfigurability and high performance make them very interesting devices for space applications, where design changes are usually not possible once the mission has started. The reconfigurability of FPGAs allows to remotely correct possible problems in the design, but also to add or modify functionalities to the circuits implemented in the system.

The effectiveness of the hardening techniques developed and implemented in FPGAs has been proven by fault injection experiments and by testing in particle accelerator facilities prepared for irradiation of electronic devices.

Irradiation tests are the industry standard for testing the behavior of all electronic systems that are sent on a space mission. The purpose of these tests is to emulate in an accelerated manner the radiation conditions to which the systems will be subjected once they have been launched and to determine their resistance to TID, DD and/or SEEs. Depending on the effects to be observed, the particles chosen for the radiation vary, being able to choose between electrons, neutrons, protons, heavy ions, photons... In particular, the radiation tests carried out in this work, being a study of hardening techniques for digital electronic systems, are aimed at establishing the sensitivity of the circuits studied against a type of SEE known as Single Event Upset (SEU), in which radiation modifies the logic value of a memory element. For this purpose, we have used radiation experiments with protons at the Centro Nacional de Aceleradores (CNA, Spain), the Paul Scherrer Institut (PSI, Switzerland) and radiation experiments with neutrons at the ISIS Neutron and Muon Source laboratory (ChipIR, UK).

The sensitivity of a circuit is usually measured in terms of its cross section with respect to a given particle, calculated as the ratio between the number of faults found in the test and the amount of ionizing particles per area used in the radiation campaign. This metric is used to estimate the number of faults that will be caused by the radiation over the lifetime of the system, but also to gather data about the effectiveness of the implemented hardening systems and to help improve them.

The fault injection method used in this Thesis as a complement to radiation is based on modifying the logic value of the data stored in the configuration memory of the FPGA. This memory stores the description of the operation of the circuit implemented in the FPGA, so modifying its values is equivalent to modifying the circuit. In FPGAs that use SRAM technology in their configuration memories, such as those used in this Thesis, this is the component most sensitive to radiation, so it is possible to compare the results of fault injection and radiation campaigns. Analogously to the cross section, in fault injection experiments we can talk about the error rate, calculated as the ratio between the number of faults found and the amount of memory bits injected.

Throughout this Thesis, different circuits hardened by Triple Modular Redundancy have been developed and their performance has been compared with those of other Approximate Redundancy techniques, namely Reduced Precision Redundancy (RPR), Reduced Resolution Redundancy (RRR) and Optimized Redundancy for Composite Algorithms (ORCA). The latter two are original contributions presented in this Thesis.

- Reduced Precision Redundancy is based on the use of two redundant replicas that compute results with a smaller number of bits than the original circuit. For each output data, the result of the original circuit and the two reduced-precision results are compared. If the two reduced-precision results are identical and their difference with the full-precision result is greater than a certain threshold value, a fault is considered to exist in the original circuit and the reduced-precision result is used to correct it. In any other case, the original result is considered correct, even though it may contain tolerable errors below the comparison threshold. Compared to a TMR-hardened circuit, RPR designs use fewer resources, due to the reduced precision of redundant circuit calculations. However, this also affects the quality of the results obtained when an error is corrected.

  In this work we also explore Scaled RPR as a method of obtaining an optimal balance between accuracy and resource consumption. In this variant of the RPR technique, the results of each computational stage in the redundant circuits have a different precision, increasing towards the last stages of the pipeline, where the result has the same accuracy as the original circuit. This method increases the quality of the corrected data while reducing the resources used for hardening.

  The results of the radiation and fault injection campaigns performed on the RPR hardened designs suggest that the reduction of resources is not only beneficial in terms of resources and energy used by the system, but also leads to a reduction of the sensitivity of the circuits, measured both in cross section and error rate.

- Reduced Resolution Redundancy is a technique originally proposed in this Thesis. It is suitable for algorithms that work with information in the form of packets whose individual data have some relationship with each other, such as an image processing algorithm. In the RRR technique, two redundant circuits are added that compute the results with a fraction of the original input data. After calculation, the decimated results

can be interpolated to obtain an approximate result of the same size as the result of the original circuit. Once interpolated, the results of the three circuits can be compared to detect and correct faults in a similar way to that used in the RPR technique. By taking advantage of hardware design features, the decrease in the amount of data processed by the Reduced Resolution circuits are translated into a decrease in resources, rather than a decrease in computational time. In this way, the RRR technique is able to reduce resource consumption compared to what would be required if TMR hardening were used.

The results of the experiments performed on designs hardened with Reduced Resolution Redundancy suggest that the technique is effective in reducing the resources used and, as in the case of Reduced Precision Redundancy, its sensitivity is also reduced compared to the sensitivity of the same circuit hardened with TMR. In addition, there is a notable reduction in the sensitivity of the circuits to uncorrectable errors, compared to the same result in TMR and RPR. This type of error encompasses those produced by failures in the comparison and voting logic or those in which a single SEU produces failures in the results of two or more of the redundant circuits at the same time, which is known as Common Mode Failure (CMF). However, it is also observed that the quality of the corrections performed using this method worsens slightly.

- Optimized Redundancy for Composite Algorithms is also an original contribution of this Thesis. It is suitable for algorithms whose final result can be expressed as the composition of intermediate results computed in previous stages. To harden a circuit using this technique, two redundant circuits are added that are different from each other, each of them processing a different part of the input data set. Each of these approximate circuits computes an intermediate result. The composition of the two intermediate results gives a result identical to that of the original circuit in the absence of faults.

  Fault detection is performed by comparing the result of the original circuit with that of the composition of the approximate circuits. In case they are different, the origin of the fault can be determined by comparing the intermediate approximate results against a threshold. If the difference between the intermediate results exceeds the threshold, it means that the fault has occurred in one of the approximate circuits and that the result of the composition should not be used in the output. As with Reduced Precision Redundancy and Reduced Resolution Redundancy, using a comparison threshold implies the existence of tolerable errors. However, this hardening technique allows precise, rather than approximate, corrections to be made in most cases, which improves the quality of the results with respect to other approximate hardening techniques, while reducing the resources used by the hardened system compared to traditional techniques.

  The results of experiments performed with hardened designs using Optimized Redundancy for Composite Algorithms confirm that this hardening technique is capable of producing precise corrections in a high percentage of the events. Its sensitivity to all types of errors and to uncorrectable errors is also diminished compared to that obtained with Triple Modular Redundancy.

The results presented in this Thesis support the idea that Approximate Redundancy techniques are viable alternatives to the usual radiation hardening techniques, provided that the system characteristics tolerate the presence of inaccurate results. Approximate Redundancy manages to reduce the area and resource consumption of hardened circuits at the cost of making approximate corrections, ranging from obtaining lower accuracy in the corrected results (RPR), corrections based on close correct results (RRR) or exact corrections in some cases and lower precision in others (ORCA). In either case, these approximate corrections may be perceived as mere noise by the system and not affect its functionality.

Radiation and fault injection campaigns show that Approximate Hardening techniques reduce the sensitivity of hardened circuits (with respect to that of the same circuit hardened with Triple Modular Redundancy) as a consequence of reduced resources. Comparing the results of both types of campaigns has allowed us to establish a relationship between them, which validates the fault injection method and positions it as a fast and inexpensive way to obtain preliminary results. Fault injection has allowed us, on the one hand, to refine the performance of the proposed techniques without the need to test them in radiation campaigns or long simulation campaigns, and on the other hand, to know in advance the sensitivity and error correction capacity of the proposed hardening techniques. These two advantages have allowed us to carry out a more adequate planning of the radiation campaigns, making the best use of the available radiation time and minimizing the costs of the experiments.

In addition to these features, the results obtained suggest that the way in which the techniques of Reduced Resolution Redundancy and Optimized Redundancy for Composite Algorithms treat the data reduces the incidence of Common Mode Failures, which are responsible for a large part of the uncorrectable failures in all the systems hardened by RHBD techniques.

LIST OF ACRONYMS

A

ABFT Algorithm-Based Fault Tolerance. 69

ALD Area Limited Design. 99

APSoC All-Programmable System on Chip. 71, 84, 101, 123, 132, 136, 159

ARM Advanced RISC Machine. 71, 84, 101, 123, 132, 136

ASIC Application-Specific Integrated Circuit. 17, 19, 37, 59, 69, 96, 169

ATMR Approximate Triple Modular Redundancy. 49, 79, 80, 96, 111–113

B

BOX Buried OXide. 37

BTMR Block Triple Modular Redundancy. 48, 49, 59, 67, 78, 81, 95

BU Butterfly Unit. 70, 98, 116

C

CLB Configurable Logic Block. 83

CME Coronal Mass Ejection. 8

CMF Common-Mode Failures. xv, xxi, 19, 60, 68, 69, 71–75, 78, 79, 86, 87, 90–92, 96, 102–104, 106, 121, 123, 125–128, 135, 144, 145, 151, 156, 158, 161, 162, 164, 169–171

CMOS Complementary Metal-Oxide Semiconductor. 13, 16, 37–39

CNA Centro Nacional de Aceleradores. ix, xi, xiv, xix, 71, 84, 101, 136, 190

COTS Commercial Off-The-Shelf. x, xiii, xviii, 2, 22, 28, 29, 57, 61, 65, 66, 68, 111, 128, 150, 151

CRAND Cosmic Ray Albedo Neutron Decay. 10

D

DD Displacement Damage. vii, x, xii, xiii, xviii, xix, 13, 28, 30, 37

DDD Displacement Damage Dose. 13

DDR Design Diversity Redundancy. 152

DICE Dual Interlocked CEll. 54

DMA Direct Memory Access. 141, 143, 145, 146, 194

DMR Dual Modular Redundancy. 44, 50

DSP Digital Signal Processing. 3, 49, 67–69, 80, 81, 83, 111, 112, 114, 117, 118, 124, 126, 131–135, 157, 160, 163, 166–168

DT2 Dual Duplex Tolerant to Transients. 66

DTMR Distributed Triple Modular Redundancy. 46–49, 59, 60, 81

DWC Duplication With Comparison. viii, xi, xiii, xix, 44, 50, 112, 166

E

ECC Error-Correcting Code. 66

EDAC Error Detection And Correction. 51, 54, 56, 96, 111, 171

ELDRS Enhanced Low Dose Rate Sensitivity. 12

ELT Enclosed Layout Transistor. 36

ESA European Space Agency. 3, 130, 132, 140

ESCC European Space Components Coordination. 30

F

FF Flip-Flop. 86, 117, 118, 124, 134, 167

FFT Fast Fourier Transform. 50, 68–73, 75, 76, 78, 79, 81–89, 92, 95, 96, 98–105, 107, 108, 111–113, 115–120, 122–127, 130, 132–140, 142, 143, 145, 150, 154–157, 159–162, 167, 168

FIFO First-In First-Out. 70, 98, 115–118, 137, 194

FIR Finite Impulse Response. 79, 81, 113, 150, 154, 157–159, 163, 164

FIT Failure In Time. 27

FMEA Failure Mode and Effect Analysis. 24

FP Full Precision. 81–83, 85–89, 97, 100, 112, 117, 118, 151

FPGA Field-Programmable Gate Array. viii, xi, xiii, xiv, xix, xx, 2, 3, 5, 17, 19, 37, 43, 46, 48, 51, 56–62, 64–69, 75, 78–81, 84, 96, 97, 101, 108, 117, 121, 123, 125, 127, 130, 131, 134, 136, 144, 145, 150, 151, 156, 157, 159, 161, 165, 167, 169, 171, 190, 191, 193–195

FR Full Resolution. 114, 115, 118–122, 126, 133, 135, 137, 138, 141

FSM Finite State Machine. 51

G

GCR Galactic Cosmic Radiation. 10

GEO Geostationary Earth Orbit. 9, 12, 23

GNSS Global Navigation Satellite Systems. 21

GPIO General Purpose Input/Output. 192

GTMR Global Triple Modular Redundancy. 47

H

HDL Hardware Description Language. 58

HIT Heavy-Ion Tolerant. 54

HLS High-Level Synthesis. 132, 141

I

IEEE Institute of Electrical and Electronic Engineers. 130, 150

IP Intellectual Property. 48, 70, 71, 79, 82, 84, 89, 92, 101, 123, 136, 143, 157, 159, 160, 190–195

ISS International Space Station. 9

K

kerma Kinetic Energy Released to Matter. 13

L

LCL Latching Current Limiter. 39, 40, 65

LEO Low Earth Orbit. 9, 10, 23

LET Linear Energy Transfer. 14, 25

LTMR Local Triple Modular Redundancy. 45–47

LUT Look-Up Table. 83, 84, 86, 90, 100, 107, 108, 117, 124, 126, 134, 142, 163, 167, 168

LUTRAM Look-Up Table Random Access Memory. 83

M

MBU Multiple Bit Upset. 19, 52, 56

MCU Multiple Cell Upset. 19, 69, 102

MDC Multi-path Delay Commutator. 70

MEO Medium Earth Orbit. 9

MOSFET Metal-Oxide-Semiconductor Field-Effect Transistor. 17, 32, 39

MPSoC MultiProcessor System-On-Chip. 2, 3, 57, 65, 66, 130, 171, 190

MSE Mean Squared Error. 75, 87, 103, 139

MTBF Mean Time Between Failures. 27

MTTF Mean Time To Failure. 27

N

NASA National Aeronautics and Space Administration. 21

NIEL Non Ionizing Energy Loss. 13

NIR Near InfraRed. 132, 135, 136, 140–146

NMOS Negative-channel Metal Oxide Semiconductor. 16, 36, 38

NMR N-Modular Redundancy. 44, 45, 49, 52, 62

NNI Nearest Neighbour Interpolation. 133, 141

O

ORCA Optimized Redundancy for Composite Algorithms. viii, xi, xiv, xvi, xx, xxii, xxxvi, 150, 152–156, 158–165, 168–170

P

PC Program Counter. 65

PIPB Propagation-Induced Pulse Broadening. 19

PL Programmable Logic. 132

PMOS Positive-channel Metal Oxide Semiconductor. 16, 38, 43

POA Post Oxidation Annealing. 36

PS Processing System. 132

PSNR Peak Signal-to-Noise Ratio. 68, 75, 87, 88, 91, 103, 104, 106–108, 125, 126, 139, 140, 143, 144, 161, 162, 170

R

R2^{2}SDF Radix-2^2 Singlepath-Delay Feedback. 98, 99, 102

R2SDF Radix-2 Singlepath-Delay Feedback. 70, 98, 99, 102, 115–117, 123, 156

R4SDF Radix-4 Singlepath-Delay Feedback. 70, 98, 99

RAM Random Access Memory. 52, 58, 83, 100

RHBD Radiation Hardening By Design. vii, viii, x–xiii, xvii–xix, xxii, 22, 37, 39, 61, 131, 151, 171

RHBP Radiation Hardening By Process. vii, x, xii, xviii, 22, 35, 37, 40, 61, 131

RoRa Reliability-oriented place and Route Algorithm. 60

RP Reduced Precision. 81–83, 85–89, 91, 92, 97, 100, 103, 107, 112, 117, 118

RP-TMR Reduced Precision Triple Modular Redundancy. 79, 81, 82, 84, 92

RPR Reduced Precision Redundancy. viii, xi, xiv–xvi, xx–xxii, 4, 49, 50, 79, 81–83, 85–89, 91, 92, 96–108, 111–114, 117–119, 122–124, 126, 127, 131, 132, 135, 137–140, 147, 151–153, 160–164, 166, 167, 170

RR Reduced Resolution. xxxv, 114, 115, 117–122, 127, 133, 135, 137, 138, 141

RRR Reduced Resolution Redundancy. viii, xi, xiv, xvi, xx–xxii, xxxv, 113–115, 117, 118, 122–127, 130, 132–147, 152, 156, 160–162, 167–170

RTS Random Telegraph Signal. 13

S

SAA South Atlantic Anomaly. 10

SDC Single-path Delay Commutator. 70

SDF Single-path Delay Feedback. 70, 98, 115–117, 122

SEB Single Event Burnout. 17

SECDEC Single Error Correction - Double Error Correction. 55

SEDR Single Event Dielectric Rupture. 17

SEE Single Event Effects. vii, x, xii, xiii, xviii, xix, 14, 25, 31–35, 58, 65, 96

SEFI Single Event Functional Interrupt. 18–20, 64

SEGR Single Event Gate Rupture. 17

SEHE Single Event Hard Error. 17

SEL Single Event Latch-up. 16, 37, 64

SEM Soft Error Mitigation. 71, 84, 89, 101, 143, 193–195

SEP Solar Energetic Particles. 8

SESB Single Event Snapback. 16

SET Single Event Transient. 18, 19, 34, 38, 40–43, 50, 52, 96

SEU Single Event Upset. xiv, xv, 18, 19, 33, 34, 38, 43, 46–48, 51, 53, 56, 58–60, 64, 69, 96, 97

SoC System On Chip. 71, 84, 101, 123

SOI Silicon On Insulator. 37, 38, 40

SPA Single Photon Absorption. 34

SPENVIS SPace ENVironment Information System. 28

SRAM Static Random Access Memory. xiv, 5, 19, 20, 52, 58, 59, 61, 64, 69, 75, 80, 81, 96, 156

STI Shallow Trench Isolation. 36

T

TID Total Ionizing Dose. vii, x, xii, xiii, xviii, xix, 11–13, 28, 30, 35–38, 40, 62

TMR Triple Modular Redundancy. viii, xi, xiii–xv, xix–xxi, 44, 45, 48–51, 59–62, 67–69, 71, 79–83, 85–89, 91, 92, 95–98, 100–106, 108, 111–113, 117, 118, 123, 124, 126, 127, 133, 137–139, 142, 146, 150–152, 156, 159–164, 166, 167, 169, 170, 194

TPA Two Photon Absorption. 34

U

UART Universal Asynchronous Receiver-Transmitter. 71, 101, 123, 191, 194, 195

USA United States of America. 21

USB Universal Serial Bus. 191

USSR Union of Sovietic Socialist Republics. 21

V

VHDL Very High Speed Integrated Circuits Hardware Description Language. 51, 58

CONTENTS

ACKNOWLEDGEMENTS . . . iii
PUBLISHED AND SUBMITTED CONTENT . . . iv
OTHER RESEARCH MERITS . . . vi
RESUMEN NO TÉCNICO . . . vii
NON-TECHNICAL SUMMARY . . . x
RESUMEN TÉCNICO . . . xii
TECHNICAL SUMMARY . . . xviii
LIST OF ACRONYMS . . . xxiii
ACRONYMS . . . xxviii
1. INTRODUCTION . . . 1
1.1. Problem statement . . . 1
1.2. Aim and scope . . . 2
1.3. Methodology . . . 3
1.4. Significance . . . 4
1.5. Funding . . . 5
1.6. Document overview . . . 5
2. STATE OF THE ART . . . 7
2.1. Ionizing radiation in the space environment . . . 7
2.1.A. Sources of radiation . . . 7
2.2. Effects on electronics . . . 11
2.2.A. Total Ionizing Dose . . . 11
2.2.B. Displacement Damage . . . 13
2.2.C. Single Event Effects . . . 14
2.3. Space Industry . . . 21
2.4. Dependability and qualification . . . 23
2.4.A. Dependability . . . 23
2.4.B. Radiation Hardness Assurance: Qualification . . . 28

2.5. Hardening against radiation . . . 35
2.5.A. TID hardening . . . 35
2.5.B. DD hardening . . . 37
2.5.C. SEL hardening . . . 37
2.5.D. SET hardening . . . 40
2.5.E. SEU hardening . . . 43
2.6. Hardening against SEUs . . . 43
2.6.A. Digital circuits . . . 43
2.6.B. Memory elements . . . 52
2.6.C. Hardening complex devices . . . 57
2.7. Final considerations . . . 66
3. ERROR SENSITIVITY STUDY OF FFT ARCHITECTURES IMPLEMENTED IN FPGA . . . 67
3.1. Introduction . . . 67
3.2. Related work . . . 69
3.3. FFT architectures under study . . . 69
3.4. Experimental results . . . 71
3.5. Conclusions . . . 75
4. ANALYZING REDUCED PRECISION REDUNDANCY UNDER PROTON IRRADIATION . . . 78
4.1. Introduction . . . 78
4.2. Related work . . . 80
4.3. Reduced Precision Redundancy . . . 81
4.4. Experimental setup . . . 84
4.5. Experimental results . . . 85
4.6. Conclusions . . . 92
5. ANALYZING SCALED REDUCED PRECISION REDUNDANCY FOR ERROR MITIGATION UNDER PROTON IRRADIATION . . . 95
5.1. Introduction . . . 95
5.2. Scaled FFT architectures . . . 97
5.3. Experimental setup . . . 100
5.4. Experimental results . . . 102

5.5. Conclusions 108

6. REDUCED RESOLUTION REDUNDANCY: A NOVEL APPROXIMATE ERROR MITIGATION TECHNIQUE 110

6.1. Introduction 110

6.2. Background and related work 112

6.3. Reduced Resolution Redundancy 113

6.4. Application to FFT case study 115

6.5. Experimental setup 122

6.6. Results 123

6.7. Conclusions 127

7. EVALUATING REDUCED RESOLUTION REDUNDANCY FOR RADIATION HARDENING 130

7.1. Introduction 131

7.2. Reduced Resolution Redundancy 132

7.3. Experimental setup 136

7.4. Experimental results 136

7.4.A. FFT benchmark 137

7.4.B. NIR HAWAII image processing benchmark 140

7.5. Conclusions 146

7.6. Acknowledgments 147

8. OPTIMIZED REDUNDANCY FOR COMPOSITE ALGORITHMS 150

8.1. Introduction 150

8.2. Background and related work 150

8.3. Optimized Redundancy for Composite Algorithms (ORCA) 152

8.4. The FFT and FIR filter case studies 154

8.4.A. FFT benchmark 154

8.4.B. FIR filter benchmark 157

8.5. Experimental setup 159

8.6. Experimental results 160

8.6.A. FFT benchmark 160

8.6.B. FIR filter benchmark 163

8.7. Conclusions 165
9. CONCLUSIONS AND FUTURE RESEARCH 166
BIBLIOGRAPHY 172
A. MATERIALS AND METHODS 190
A.1. Resources 190
A.1.A. Devices Under Test 190
A.1.B. Radiation facilities 190
A.1.C. Experimental setup 191
A.2. Methodology 193
A.2.A. Fault injection 193
A.2.B. Data acquisition 194
A.2.C. Data processing 195

LIST OF FIGURES

2.1 Space radiation environment. 8
2.2 Van Allen radiation belts. 9
2.3 Electron-hole pair creating and hole trapping. 12
2.4 Charge collection process in a reverse-biased junction. 15
2.5 Latch-up path between the PMOS and NMOS transistors in the CMOS technology. 16
2.6 Single Event Snapback effect in a MOS transistor. 16
2.7 Parasitic structure of Single Event Burnout in MOSFET. 17
2.8 Single Event Gate Rupture effect in a power MOSFET structure. 17
2.9 Examples of Single Event Upsets in an SRAM memory. 20
2.10 The Bathtub curve. 25
2.11 a 26
2.11 Cross section vs. LET curves in an SRAM memory. 26
2.12 Enclosed Layout Transistor design. 36
2.13 CMOS transistor protected with guard rings. 37
2.14 NMOS transistor hardened using SOI. 38
2.15 CMOS transistor hardened using Triple Wells. 38
2.16 Buried layer in an NMOS transistor. 39
2.17 CMOS transistor hardened against SEL using a p epitaxial layer. 39
2.18 Basic configuration of a Latching Current Limiter. 40
2.19 Analog redundancy architecture. 42
2.20 Basic structure of the DWC hardening method. 44
2.21 N-Modular Redundancy basic schematic. 45
2.22 Block diagram of the LTMR mitigation technique. 46
2.23 Block diagram of the DTMR mitigation technique. 47
2.24 Block diagram of the GTMR mitigation technique. 47
2.25 Block diagram of the BTMR mitigation technique. 48
2.26 SET filtering TMR. 51

2.27 6T SRAM cell. 53
2.28 Resistive and capacitive mitigation techniques for memory cells. 53
2.29 Dual Interlocked Cell (DICE) architecture. 54
2.30 Interleaved SRAM memory. 57
2.31 Resource placement for the same design hardened using DTMR and BTMR. 59

3.1 Pipelined FFT architectures. 71
3.2 Error rates and CMF rates of the tested architectures. 73
3.3 LUT as logic utilization vs. error rate. 73
3.4 Error classification of the words in faulty frames. 74

4.1 Architecture of an RP-TMR design to harden an FFT. 82
4.2 Utilization of resources of the tested benchmarks. 84
4.3 Error classification of the faulty frames of the radiation experiments. . . . 86
4.4 Error classification of the faulty frames of the injection experiments. . . . 90
4.5 Correlation between the Error rate and the Cross-section. 91

5.1 Block diagram of an RPR hardening method. 97
5.2 Stages in FFT architectures. 99
5.3 Scaling pattern in FFT pipelined architectures. 99
5.4 Resource utilization of the stages of full precision, reduced precision and scaled reduced precision Radix-2^2 FFTs. 100
5.5 Resource utilization of the different designs. 101
5.6 Error classification of the faulty words found in the radiation experiments. 103
5.7 Correlation between the Error rate and the usage of LUTs as logic. 106
5.8 Correlation between the Cross section and the Error rate. 106
5.9 Error classification of the faulty words found in the frames. 107

6.1 Block diagram of an RPR system to correct errors in FFT blocks. 113
6.2 Block diagram of the proposed Reduced Resolution Redundancy technique. 115
6.3 Architecture of a Radix-2 SDF pipelined FFT. 116
6.4 Architecture of the proposed R2SER2SDF pipelined FFT. 117
6.5 Utilization of resources of FFTs with different precisions and resolutions. 118

6.6 Comparison of the outputs of the FR FFT and the RR FFT after the conditioning step. 119
6.7 Proposed RRR voting algorithm for error correction. 120
6.8 Structure of the proposed voting algorithm for the RRR hardening technique. 121
6.9 Behavior of the voting algorithm. 122
6.10 Utilization of resources of the different designs for increasing data precisions. 125

7.1 Block design of an FFT hardened with the RRR technique. 134
7.2 Block diagram of an image processing algorithm hardened with the RRR hardening. 134
7.3 Utilization of hardware resources of Full and Reduced Resolution individual implementations of the FFT and NIR HAWAII algorithms. . . 135
7.4 Diagram of a basic RRR voting logic. 136
7.5 Pipeline stages of (a) a R2 FFT implementation and (b) a R2SER2 FFT architecture. 137
7.6 Resource consumption of different hardening techniques applied to the FFT benchmark. 138
7.7 Error classification of the faulty words in the FFT benchmarks. 140
7.8 Post synthesis resource consumption of the TMR and RRR hardening techniques applied to the NIR HAWAII benchmark. 142
7.9 Error classification comparison between the fault injection and radiation campaigns in the RRR designs. 145
7.10 Error classification of the NIR HAWAII results in fault injection and neutron irradiation after removing the uncorrectable errors found. 146

8.1 Block diagram of a basic ORCA design. 153
8.2 Diagram of an 8-point FFT composed using two 4-point FFT blocks. . . . 155
8.3 Block diagram of an FFT benchmark hardened using the ORCA technique. 156
8.4 FIR filter decomposition in two composite blocks. 158
8.5 Block diagram of a FIR filter benchmark hardened with the ORCA technique. 159
8.6 Synthesis results of the ORCA FFT benchmark and comparison with other hardening techniques applied to an R2SDF FFT. 160

8.7 Error classification of the experiments conducted in the ORCA FFT benchmark and comparison with other hardening methods. 162
8.8 Synthesis results of the ORCA FIR filter benchmark and comparison with other hardening techniques applied to similar FIR filter implementations. . 163
8.9 Error classification of the experiments conducted in the ORCA FIR filter benchmark and comparison with other hardening methods. 164

9.1 a 167
9.1 Correlation between the number of LUTs employed by the designs and their sensitivity. 168
9.2 Correlation between the number of LUTs and the Cross section. 168
9.3 Correlation between the Error Rate and the Cross section. 169

A.1 Experimental setup for the irradiation and fault injection experiments. . . 192
A.2 Experimental setup. 193
A.3 Architecture of the testbench system. 195

LIST OF TABLES

2.1 RADIATION EFFECTS ON ELECTRONICS 20

2.2 Hamming parity table. 55

3.1 Resource utilization of the different architectures. 72

3.2 Results of the fault injection campaigns. 73

3.3 Error classification of the reported faults. 75

4.1 Experiments performed and their features. 83

4.2 Results of radiation experiments. 85

4.3 Classification of erroneous words in radiation experiments. 89

4.4 Results of injection campaigns. 90

4.5 Classification of erroneous words in injection campaigns. 92

5.1 Results from the irradiation experiments. 103

5.2 Results from the fault injection experimentsin Radix-2 and Radix-4 architectures. 105

5.3 Results from the fault injection experiments in Radix-2^2 architecture. . . . 105

6.1 Results from the injection experiments. 126

6.2 Error classification of the faulty words in the erroneous frames. 127

7.1 Radiation results of the RRR FFT design and comparison with an RPR FFT implementation 139

7.2 High-level synthesis timing and resources estimates for the detectCosmicRay function 141

7.3 Injection results of the RRR NIR HAWAII design and comparison with an RRR FFT implementation 143

7.4 Radiation results of the RRR NIR HAWAII design under neutron irradiation and comparison with the RRR FFT implementation 144

8.1 Results for the ORCA FFT hardening technique under injection experiments and comparison with other hardening methods. 161

8.2 Results for the ORCA FIR filter hardening technique under injection experiments and comparison with other hardening methods. 164

9.1 Overhead introduced by different hardening techniques for a Radix-2 FFT benchmark. 166

1. INTRODUCTION

Outside the protective magnetic field of the Earth, highly-energetic atomic and subatomic particles travel unhindered across space. These particles are by-products of nuclear reactions originated in stars around us that eventually reach our vicinity and are deflected or trapped by the magnetic field of the Earth.

The nature of these particles is diverse: some are electrically charged, others are neutral, they come in a large range of mass and atomic number, and their energy spectrum is also wide. What these particles have in common is that they are are able to interact with matter and produce significant effects. Energy transmitted by these particles is called radiation. Charged particles form the so-called ionizing radiation and neutral particles are part of the non-ionizing radiation.

The interaction of radiation with space and avionics systems, specially electronic systems, has been an extensively studied topic because of the catastrophic effects it may induce in systems that have long development periods and are expensive in economic terms. In recent times, technological advancements and the incursion of new actors in the space market have opened new research lines in the field to achieve higher performance and lower the costs of the missions. Radiation effects on electronic systems is also an increasing concern in other high-reliability application areas at the ground level, such as communications, high-performance computing, transportation and biomedical devices.

Protecting electronic devices against the effects of radiation is the main topic of this Thesis.

1.1. Problem statement

Traditional space development has strict protocols and methods to avoid any kind of failures during the operational time of the mission. As a result, development is long and expensive, but the resultant systems are robust and usually last longer than their expected lifetime. To ensure the robustness of the electronic devices mounted in flight models, they are either manufactured or designed using special error mitigation techniques against radiation effects.

The error mitigation techniques for radiation effects usually impose a penalty in power and resources needed in exchange for the hardness. This fact, combined with the very long time passed from part qualification and selection to the end of the mission, results in outdated electronic equipment being used in missions, compared to the electronic devices present on our daily lives.

Recent trends in space development tend towards planning shorter missions, assuming more risks to achieve lower expenses and higher computational performance. An

important tool to attain this objective is the increasing usage of Commercial Off-The-Shelf (COTS) components. COTS are components that are typically used for applications at ground level, thus, they do not have any inherent radiation hardness, although they might have been manufactured to withstand wider temperature ranges, as is the case of automotive-graded components.

Error mitigation must be implemented in COTS to correct non-destructive failures, as well as testing them against destructive failures. These techniques implemented in COTS have been proven effective and sufficient to correct errors in radiation-exposed systems, enabling higher computation performance and providing additional capabilities to space applications.

Nevertheless, the high power and resource consumption of these error mitigation techniques might be excessive for sub-systems in which a not so high dependability is acceptable. The system as a whole could benefit from having more power and resources available, while said sub-systems could be hardened using an Approximate Error Mitigation technique, that still ensures a high reliability.

Approximate Error Mitigation is a Radiation Hardening By Design concept encompassing a series of hardening techniques that strive for the reduction of the amount of resources, power or execution time needed by the error correction mechanisms, in exchange for the exactitude of the correction. In case of error, these mitigation techniques offer results that are an approximate version of the expected result.

1.2. Aim and scope

The work carried out and presented in this Thesis revolves around the development of error mitigation techniques for digital circuits, specifically for digital signal processing, that are less expensive in terms of resources and power consumption. The idea of approximate computing is explored to this end.

Because of the increasing interest of the space industry and the ease to use and procure, the proposed techniques have been implemented using COTS components. Specifically, in COTS FPGAs and MultiProcessor System-On-Chips (MPSoCs), that provide high flexibility and performance.

The error mitigation techniques have been tested under standardized methods used in the industry to qualify electronic devices that may be utilized in space missions, as part of the Radiation Hardness Assurance process.

Thus, the objectives of this work can be summarized as follows:

- Study and development of Approximate Error Mitigation techniques for digital signal processing applications, including the proposal of new techniques.
- Implementation of the developed techniques in COTS FPGAs and MPSoCs.

- Validate the performance of the developed techniques using fault injection and radiation campaigns at proper facilities.

1.3. Methodology

To attain the proposed objectives we carried out a preliminary research on three issues that would be fundamental to the rest of the study: industry-representative DSP architectures to be used as benchmarks, a versatile and easy-to-use setup to be utilized in both fault injection and irradiation experiments, and previously proposed error mitigation techniques for radiation-tolerant digital circuits.

For the results of the research to be meaningful and facilitate further research and adoption of the proposed techniques, the benchmarks used in the experiments must be representative of the designs used in real space applications, with enough complexity and, where possible, enough configuration options. For this reason we selected the Fast Fourier Transform, the Finite Impulse Response filter and an algorithm to process images from a near infrared HAWAII-2RG sensor proposed by ESA.

These algorithms, paired with the great flexibility of the FPGAs in which they have been implemented, provided us with several configuration options to test different error mitigation techniques.

The designs to be tested were implemented in the programmable logic of an MPSoC device that contains an FPGA as well as a dual-core microprocessor, which provides additional options for the acquisition and treatment of data during the experiments. This means that, besides the design under test, control and data acquisition logic can be implemented in hardware or software.

An external host computer with an ethernet connection is also key in the experimental setup for two main reasons: keeping a timestamped record of the outputs received from the benchmark and controlling different aspects of the experiment, such as automatic detection of malfunctions, turning the device on and off or sending other signals to the system under test. The ethernet connection allows for remote control and monitoring of the experiments, be it from the bunker in the irradiation facility or from home in the fault injection experiments.

Radiation and fault injection campaigns were the two methods used in this Thesis to provoke stochastic errors in the circuits under test and analyze the impact of the faults in the error mitigation techniques implemented to protect the circuit.

In radiation campaigns, the device under test is placed in the trajectory of an accelerated beam of subatomic particles, capable of causing faults in the circuit through various mechanisms. Thanks to this beam, we are able to emulate the radiation environment in which the system should work.

Irradiation campaigns are expensive and must be carefully planned so that the

experiment is carried out successfully. Moreover, the experiment is subject to the availability of the particle accelerator facility, and travel and material transportation restrictions must be observed. However, this is the standard method in the industrial and scientific space community to test the performance of electronic devices under the effects of radiation.

Fault injection, on the contrary, relies on external hardware to insert artificial faults in specific or random nodes of the circuit. Although usually not all of the resources present in the circuit may be accessible by the fault injection setup, the results may be similar to those of the irradiation campaign under certain circumstances.

Fault injection can be performed at minimal costs, which allows for longer and more accurate experiments. The main advantage of this method is being able to extensively test several benchmarks to select the most appropriate or interesting for the radiation experiments at a small money and time cost.

For the experiments carried out in this Thesis, we used the same benchmarks and setups for the fault injection and the irradiation campaigns. The advantage of using the same exact setup is that the results from different experiments may be correlated and compared to a certain extent and fault injections may be used beforehand to predict the behaviour of the experiment under the effects of radiation.

The benchmarks and experimental setup were validated using Triple Modular Redundancy, an error mitigation technique commonly used in space missions, and those results also served as a ground truth to compare the performance of the novel mitigation techniques that we propose in this Thesis.

Additionally, we also tested Reduced Precision Redundancy using the same setup under fault injection and irradiation experiments. This Approximate Error Mitigation technique was proposed previously as a way to reduce the power consumption of hardened circuits but had not been tested under radiation before.

Once the benchmarks were selected and the setup was tested under real experimental conditions, we started the development of novel Approximate Error Mitigation techniques with the idea of reducing the impact of spatial redundancy methods on digital circuits for space applications. Throughout this Thesis we will present a modification of the Reduced Precision Redundancy technique we call Scaled RPR, and two novel techniques, the Reduced Resolution Redundancy and the Optimized Redundancy for Composite Algorithms, which were validated with different benchmarks and under different irradiation and fault injection campaigns.

1.4. Significance

The results presented in this Thesis show that the proposed Approximate Error Mitigation techniques for space applications are viable alternatives to the usual hardening methods

used in traditional radiation-hardened applications.

The proposed methods boast of significant error correction capabilities, and the reduction of resources, which is desirable by itself, brings the additional benefit of reducing the sensitivity to failures of circuits hardened using Approximate Error Mitigation techniques with respect to circuits hardened using traditional techniques.

The fault injection method we have used in this Thesis to test the designs prior to irradiation has been found to be a fast and inexpensive testing method for SRAM-based FPGAs and the fault injection results have a good correspondence to the results of the irradiation campaigns performed.

1.5. Funding

This work has been supported in part by the Spanish Ministry of Science and Innovation under project PID2019-106455GB-C21, by the Community of Madrid under project no. 49.520608.9.18 and by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101008126 in the framework of the EU project RADNEXT.

1.6. Document overview

This document is organized as follows:

- Chapter 2 reviews the fundamental background of this Thesis, from the basic fault mechanisms induced by radiation to the available methods to harden electronic devices.
- Chapters 3 to 8 gather the research carried out during this Thesis and review the results obtained in the experiments to validate the hardened designs.
  - In Chapter 3 we explored different architectures of Fast Fourier Transform circuits hardened using Block Triple Modular Redundancy under fault injection experiments, finding that their error sensitivity is strongly correlated to the amount of resources used in the implementation.
  - In Chapter 4 we propose several Fast Fourier Transform serial architectures hardened using Reduced Precision Redundancy and analyze their sensitivity under proton irradiation and in fault injection experiments. The Reduced Precision Redundancy technique was tested under irradiation for the first time in this article and we could establish a correlation between the radiation and fault injection results.
  - Chapter 5 describes the Scaled Reduced Precision Redundancy technique, which we developed as an upgrade of the previously tested Reduced Precision

Redundancy technique. This approach aims at obtaining better precision in the corrected results at a small resource increase. The Scaled Reduced Precision Redundancy was evaluated with different pipelined Fast Fourier Transforms architectures under proton irradiation.

- In Chapter 6 we propose Reduced Resolution Redundancy, a novel error mitigation technique that leverages decimated versions of the input data to perform approximate correction upon error. This technique was tested using a Fast Fourier Transform as a case study and was validated with fault injection experiments, showing promising results.
- In Chapter 7 we delve further into the Reduced Resolution Redundancy technique. We provide fault injection and neutron irradiation results for the Fast Fourier Transform benchmark as well as for an image processing algorithm with a mixed software-hardware architecture. The irradiation results presented in this Chapter confirm the fault injection results.
- In Chapter 8 we present another original contribution, the Optimized Redundancy for Composite Algorithms error mitigation technique. Systems hardened with this technique are capable of performing exact corrections under certain circumstances, instead of relying on approximate results, while achieving a significant reduction of resources compared to typical hardening methods based on redundancy.

- Chapter 9 provides a conclusion to the presented work and future research lines.
- Appendix A details the materials and methods utilized in this work to develop and test the proposed radiation-hardening techniques.

2. STATE OF THE ART

This chapter provides a detailed description of the basic concepts needed to contextualize this Thesis and to allow a better understanding of the work explained in Chapters from 3 to 8. Although these chapters already include their own State of the Art sections, which provide enough information to understand them individually, the main objective of this chapter is to grant the reader a general review of the topics related to this Thesis.

In this chapter we will examine the sources of ionizing radiation and its effects on exposed electronics, how to classify the operational malfunctioning of affected devices and how the paradigm shift in the space industry has affected our view on their operation standards. Lastly, we will review the most important techniques to cope with faulty systems and how to ensure that the hardening methods applied are effective before the space system is launched.

2.1. Ionizing radiation in the space environment

Radiation is defined as the process by which a wave or a particle transfers energy through space or a material medium. Depending on the amount of transmitted energy, radiation can be classified as non-ionizing (typically less than 10 eV) and ionizing, which carries enough energy to detach electrons from atoms and molecules [9] [10].

Ionizing radiation can be dangerous for living beings, causing numerous problems in tissues and organs with long exposures. There are various sources of radiation in the Earth's surface, such as spontaneous radioactivity from materials or exposure to medical treatments. However, the main sources of ionizing radiation are located outside the planet and the Earth's atmosphere and magnetosphere are capable of stopping most of it. High altitude flights, artificial satellites and interplanetary missions are subject to diverse forms of radiation because the protective effect of the atmosphere does not affect them anymore.

As is the case with natural tissues, most materials suffer some sort of degradation of their electrical, optical and mechanical properties by being exposed to ionizing radiation over time. This degradation causes the premature malfunctioning of the endangered devices in an environment where reparations are difficult, if not completely impossible.

2.1.A. Sources of radiation

Aerospace systems are vulnerable to radiation from several origins. The type of particles, their energies and behaviour greatly differ depending on the environment in which the device is orbiting or traveling. Fig. 2.1 presents the flux density of each kind of radiation

particle that can be found in space versus their energies.

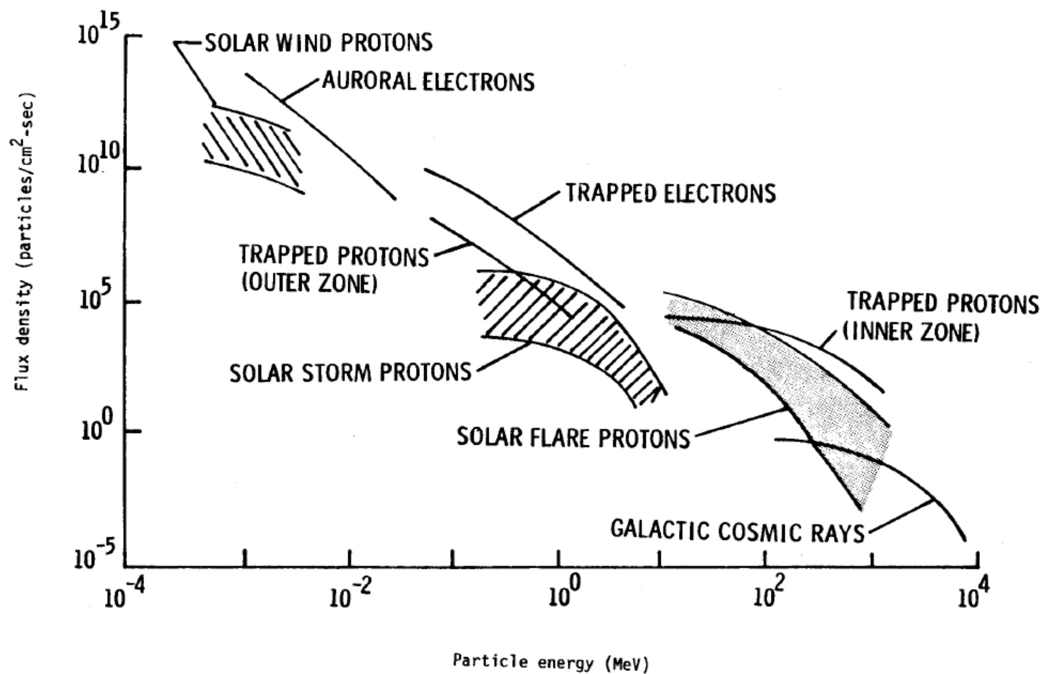


Figure 2.1: Space radiation environment [11].

Solar Wind

Solar Wind is a stream of charged particles constantly leaving the Sun in all directions. Solar Wind is composed of different kinds of particles, mostly electrons, protons and alpha particles with low energies. This stream of particles is originated in the outermost part of the atmosphere of the Sun, called the corona, when the particles are expelled by open-looped magnetic flux of the star [12].

Solar Energetic Particles

Besides the emission of Solar Wind, the magnetic flux of the Sun is also responsible of other phenomena occurring in the corona layer of its atmosphere. Regions where the solar activity is most intense, concentrates the majority of these events. Closed loops of high intensity magnetic flux in the corona create three events related to Solar Energetic Particles (SEP): Sunspots, Solar Flares and Coronal Mass Ejections (CME).

Sunspots, are regions of the photosphere with relatively lower temperatures originated when the magnetic flux is so concentrated that thermal convection no longer works. The number of yearly Sunspots is an indicative of the solar activity. The activity of the Sun changes throughout 11-years cycles and is also related to Solar Flares and CMEs.

Solar Flares and Coronal Mass Ejection are responsible for the emission of Solar

Energetic Particles, which are mostly protons, electrons and heavy ions with energies ranging from tens of keVs to GeVs [9]. These events occur in the surroundings of sunspots, when two oppositely directed magnetic fields collide in a process called magnetic reconnection. As an effect, plasma particles from the photosphere, chromosphere and corona are accelerated in all directions[12]. Some details behind these events are still largely unknown and, as a result, they cannot be predicted with precision [13] [14].

These episodes reach the Earth as the so-called Solar storms, which can affect not only the devices outside the protective atmosphere, but also cause communication and power grid interrupts in the surface.

Radiation Belts

Rotating molten materials in the outer-core of the Earth create a magnetic field around the planet. This geomagnetic field is responsible for deflecting and trapping the particles from the Solar Winds and Storms that would otherwise reach the surface and even destroy the atmosphere [15].

James Van Allen discovered in 1958 that the trapped particles were distributed in two belts that surround the Earth, named after him. The inner belt, that ranges from an altitude of about 1600 km to 13000 km, contains mainly protons of energies between 10 and 100 MeVs and a small fraction of low energy electrons (around 100 keVs). Spacecrafts in Low Earth Orbit (LEO), such as the International Space Station (ISS), are situated below the limit of the inner Van Allen belt, avoiding its effects. The outer belt, that ranges from an altitude of about 19000 km to 40000 km, contains mainly electrons of up to 10 MeVs and a small fraction of other ions. Satellites in the Medium Earth Orbit (MEO) and Geostationary Earth Orbit (GEO) traverse the outer belt in their orbits, being subject to the effects of said particles. The Van Allen belts are displayed in Fig. 2.2.

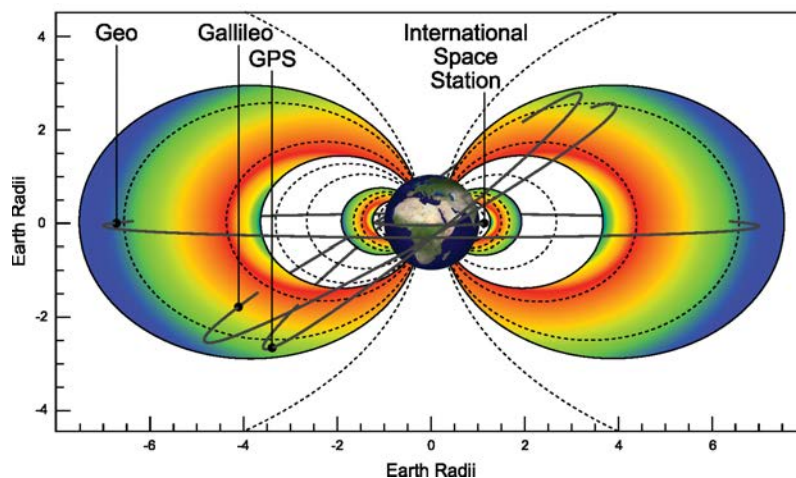


Figure 2.2: Van Allen radiation belts [16].

Due to the misalignment of the Earth rotational axis and the geomagnetic axis, there are regions of the planet where the inner Van Allen belt is closer to the surface, down to around 200 km altitude. This circumstance leads to a higher amount of trapped particles in the South Atlantic, which is known as the South Atlantic Anomaly (SAA). As a result, low-orbit satellites that cross this region are more exposed to radiation.

Galactic Cosmic Radiation

Galactic Cosmic Radiation (GCR) is a low flux of high-energy particles originated from our Sun or outside the Solar system that eventually reaches our region. The genesis of the particles coming from outside our system is still speculated, although the evidence points towards supernova explosions and active galactic nuclei [17][18].

GCRs is mainly composed of protons and fully-ionized particles ranging from helium to uranium that travel at speeds near the speed of light. Although hydrogen and helium nuclei are the most abundant in the composition of the flux, heavier (HZE, High Atomic number and Energy) particles are more hazardous for both humans and systems because of the higher energies they carry [19]. The energies of the detected Cosmic Rays reach up to a few hundreds GeVs.

The flux of Galactic Cosmic Rays is isotropic, meaning that it is uniform in all directions, and it is modulated by the solar activity. Solar wind is capable of deflecting GCRs in regions of the solar system where it still moves at supersonic speeds. As the intensity of solar winds is related to the solar activity, the flux of Galactic Cosmic Rays has an anti-correlation with the solar activity.

Neutrons

There are two main sources of neutrons. The first is the Sun, which expels neutrons as a by-product of nuclear fusion and fission reactions. These particles decay rapidly in the interplanetary region and only a small fraction reaches the Earth, but they must be taken into account for missions near the Sun [9].

The second source of neutrons is the particle cascade that occurs when a Galactic Cosmic Ray reaches the top layers of the atmosphere, colliding against an atom in a process called spallation and producing secondary particles, which, in turn, produce other secondaries. Some of these particles reach the surface of the Earth, and they pose a risk against devices at high altitudes and even at sea-level, but others are reflected back outside the atmosphere, as part of the Cosmic Ray Albedo Neutron Decay (CRAND). Those reflected particles (neutrons, electrons and protons) accumulate at the inner rim of the inner Van Allen belt and may affect devices orbiting in LEO orbits [20][21][22]. Neutrons are the main concern in high-reliability applications in the terrestrial environment, specially for equipment in high- altitude flights as their flux increases in the superior

layers of the atmosphere and near the magnetic poles of the Earth.

Secondary radiation

The spallation process that happens when a Cosmic Ray reaches the atmosphere may also happen when an energetic particle interacts with materials of the spacecraft. This secondary radiation is important for human missions and also generates background noise in sensitive detection systems [9].

2.2. Effects on electronics

Ionizing particles originated in the previously described sources interact with the electrons of the materials they traverse, transferring part of its energy to the electrons. If the energy is high enough, the electron abandons the valence band, crossing the energy gap, and reaching the conduction band. This process creates two energy carriers: the electron, and the hole left by it in the valence band of the atom. The freed electrons in the conduction band tend to move to the conduction band of other atoms, finally recombining with a hole in other valence band. This recombination effect is responsible for the generation of electric currents in semiconductor materials and charge trapping in dielectrics, which are the two main mechanisms through which electronic devices are affected. A third mechanism occurs in non-ionizing interactions, when defects are created by collisions that push atoms from their place in the crystalline matrix.

2.2.A. Total Ionizing Dose

Total Ionizing Dose (TID) is described as a cumulative damage inflicted by ionizing radiation in various materials and through various mechanisms. Although TID is also responsible for changes in the optical and mechanical properties of materials such as glass or polymers [23] [24], the most relevant TID damage in the context of this thesis is the one related to the electrical properties of electronic devices.

While metal and semiconductors are immune to TID effects, the insulation layers of electronic devices are prone to trap electron holes. When an ionized particle creates an electron-hole pair in an oxide insulator (mainly SiO_2, oxynitrides or HfO_2) the electron is able to recombine very fast outside the oxide, but the hole diffuses slower towards the Si/SiO_2 interface. Defects in the crystalline matrix trap some of these holes, creating oxide traps, while the rest of the holes advance towards the insulation interface accumulating in interface traps [25]. This process is illustrated in Fig. 2.3. It is worth noting that all trapped holes eventually recombine, but it may take a long period of time, ranging from a few minutes to years.

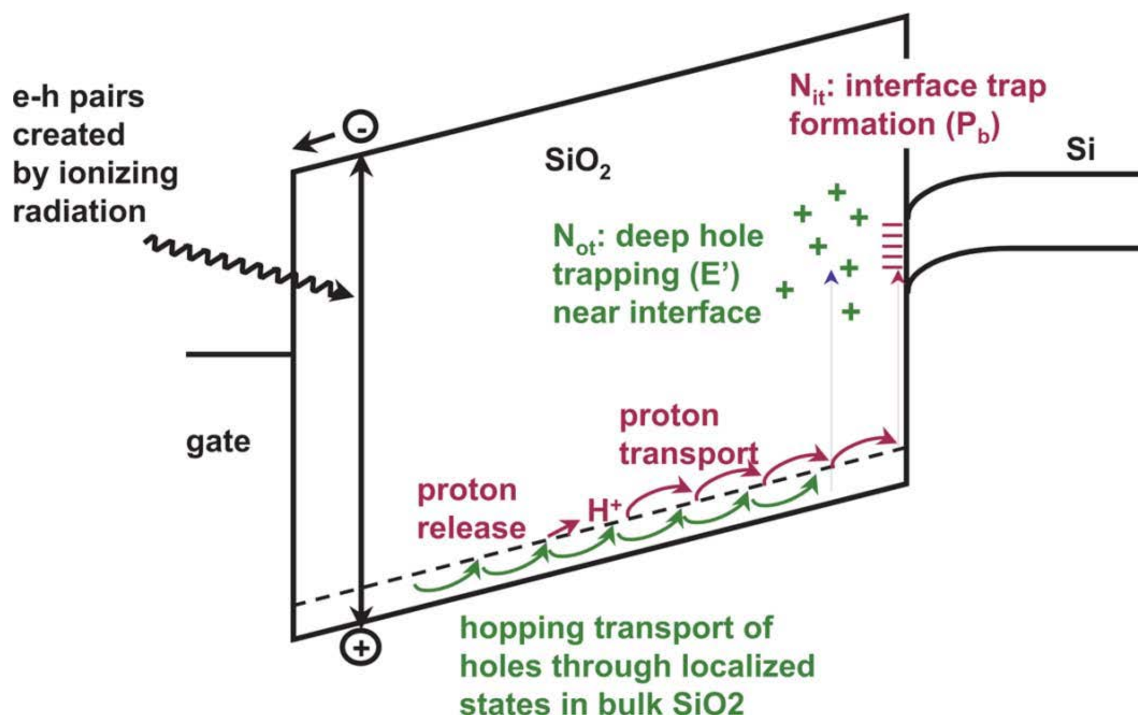


Figure 2.3: Electron-hole pair creating and hole trapping [25].

The presence of trapped holes in the transistors that form our electronic devices degrades their performance by increasing leakage currents, altering their gain or modifying the expected response to the point of rendering them useless at high doses [25].

Not only the accumulated dose is important in the case of space-related devices, but also the rate at which the dose is accumulated. Low dose rates are proved to be even more damaging to bipolar devices than higher dose rates. This phenomenon is known as Enhanced Low Dose Rate Sensitivity (ELDRS). This behaviour has been studied from several perspectives and may be explained by a conjunction of different mechanisms [26].

The radiation dose absorbed by a device is usually measured in grays (Gy), or in rad (radiation absorbed dose), where $1Gy = 100rad = 1J/kg$. The radiation absorbed by a device depends on the environmental conditions and the mission duration. These two parameters must be defined beforehand to ensure that the components are able to withstand the expected TID conditions [27]. Radiation hardened components commonly have rated radiation doses ranging from 100 krad to 1 Mrad [28].

TID effects are mainly caused by X and Gamma radiation, which is composed by high energy photons, by-products of collisions and deflections of electrons (bremmstrahlung). Thus, systems in the GEO orbit, where trapped electrons in the outer Van Allen belt are dominating, or in interplanetary missions, subject to solar wind, are particularly affected by TID [29].

2.2.B. Displacement Damage

Displacement Damage (DD), also known as Total Non-Ionizing Dose (TNID), is another cumulative effect suffered by devices exposed to ionizing radiation. However, this effect does not involve the electric charge of the particles, but their energy.

DD is caused by elastic and non-elastic collisions of protons and high energy electrons with the atoms in the crystalline lattice of a material. An affected atom can be displaced from its original position, creating a vacant, and become inserted between other atoms, creating an interstitial. Electrons and low-energy protons (<10 MeV in Si) produce single atomic displacements, whereas high energy protons are capable of initiating cascading reactions in which multiple vacant and interstitials are originated by the recoil atoms [30]. Around 90% of the defects created this way eventually recombine in the crystalline matrix, given enough time and temperature. This process is called annealing. The rest of the defects become stable and remain in the material [31].

The Displacement Damage Dose (DDD) received by an exposed system is a function of the energy lost by the incident particles in the collisions (Non Ionizing Energy Loss, NIEL) can be expressed as in Eq. 2.1, where NIEL is the Kinetic Energy Released to Matter (kerma) in $keVcm^2/g$ and ϕ is the fluence of the radiation as $particles/cm^{-2}$ [32].

$$\int_{E_{min}}^{E_{max}} \left(\frac{\delta\Phi}{\delta E}\right) NIEL(E)\, dE \tag{2.1}$$

The degradation caused by Displacement Damage manifests its effects differently on each device and technology, but it is specially important for optical devices, in which it produces increases of leakage currents in transistors and optical sensors, power losses in light-emitting diodes, optical fibers and solar cells. The case of image sensors is interesting because of the importance of DD in the quality of the images. Newer technologies have decreased the impact of TID on CMOS (Complementary Metal–Oxide Semiconductor) Image Sensors and the impact of DD is an important research topic. Degradation in these sensors comes in form of the increase of dark current, a leakage of the deposited charge that has to be read by the sensor, and the appearance of Random Telegraph Signals (RTS), bright pixels that turn on and off randomly [33].

As explained before, the most relevant particles that cause Displacement Damage effects are protons and high energy electrons. Systems subject to solar energetic particles coming from solar flares and coronal mass ejections are prone to suffer Displacement Damage. Heavy ions and neutrons also cause this type of defects, but their importance is relatively lower because of their smaller incidence: the accumulated damage due to Galactic Cosmic Rays or neutrons outside the atmosphere is very low compared to the dose received by protons and electrons. Thus, devices that cross the outer Van Allen radiation belt and low-orbit satellites near the inner belt or the South Atlantic Anomaly absorb Displacement Damage Dose on a regular basis. Moreover, during solar storms and high solar activity periods all systems can be affected by DD damage.

2.2.C. Single Event Effects

The term Single Event Effect (SEE) refers to a variety of malfunctions in electronic devices when a single energetic particle hits a sensitive node in the circuit. Depending on the node, the energy of the particle and other factors, the SEE may have no observable effect, cause a temporal interruption of the circuit operation, change the logic state of a memory cell or even cause permanent damage to the device.

Single Event Effects are caused when a particle deposits enough energy in the silicon volume of a transistor or other device near some of its sensitive nodes, causing a change in its status. The charge deposition is carried out through two different mechanisms: ionized particles directly depositing their charge or non-ionized particles that trigger nuclear reactions with atoms in the volume.

- **Direct Ionization:** an ionized particle passing through a semiconductor creates a path of electron-hole pairs, losing energy on the way until it finally stops. The energy that a particle loses due to ionization as it traverses the material is called Linear Energy Transfer (LET) and is calculated as in Eq. 2.2

  $$LET = \frac{dE}{dx} \tag{2.2}$$

  Although Linear Energy Transfer is expressed in Newtons in the International Unit System, the most common units used in the radiation field are $keV/\mu m$ or MeV/cm. The result of this equation can be normalized by dividing by the density of the traversed material in order to make comparisons. Direct ionization can be triggered by all sorts of charged particles, provided they travel at high enough speeds. Ions, electrons, protons, and muons are the main sources of direct ionization.

- **Indirect Ionization:** neutrons and photons cannot create direct ionization effects because they are electrically neutral particles, but when they traverse a material, they can undergo inelastic collisions with the atoms in the crystalline lattice. The secondary particles created in this collision, however, can create direct ionization trails. Contrary to the direct ionization mechanism, the particles that produce indirect ionization do not need high energies to affect electronic devices. The special case of radiation with low energy neutrons (in the range of 25 meV) is called thermal neutron irradiation.

For silicon devices, an electron-hole pair is created for every 3.6 eV lost by the incident particle. Not all of the charge generated by the electron-hole pairs can be collected by sensitive nodes, since the region where this occurs is very limited and depends on multiple factors such as the ion, its energy, the angle of incidence or the type of node [34].

Reverse-biased junctions, which are the functioning principle of diodes, are very sensitive to SEEs because of the high electric field formed by the P-N junction. This

high electric field favors the charge collection process. When a particle strikes a sensitive node, it creates an electron-hole pair trail through direct ionization (Fig. 2.4.a). If the ionized track is close to or crosses the depletion region of the junction, free carriers are collected by the electric field in between a nanosecond and tens of picoseconds, creating a high current transient. This first collection process is called drift (Fig. 2.4.b). After the drifting process, the depletion region continues collecting free carriers by diffusion until all electrons have been captured, recombined or diffused to other regions (Fig. 2.4.c).

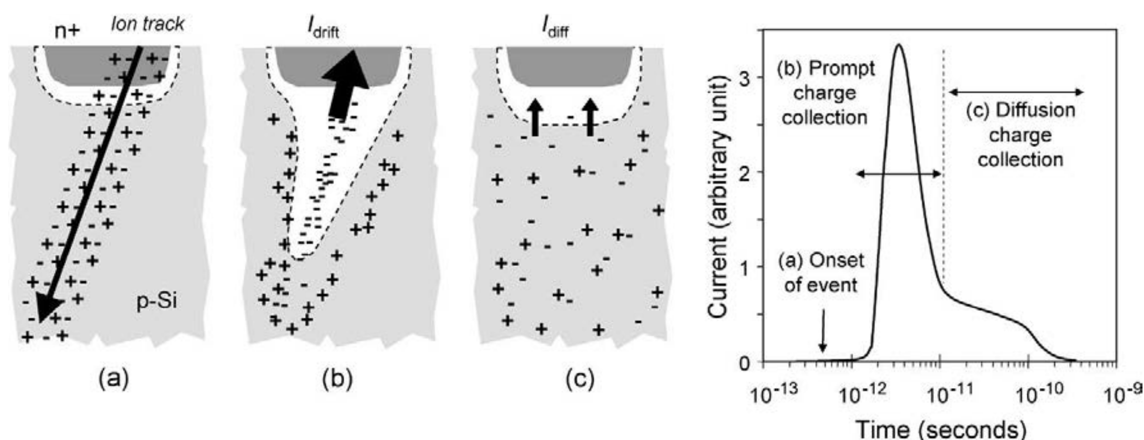


Figure 2.4: Charge collection process in a reverse-biased junction and current pulse generated in the node [34]

Every device has a critical charge, defined as the necessary charge needed to change its working state. The critical charge depends on the capacitance of the node and the operating voltage among others. Whether the collected charge surpasses this threshold and is able to affect the device in a significant way also depends on a number of factors related to the ionizing particle and physical traits of the device. It is for these reasons that this phenomenon is difficult to model and simulate.

Moreover, the ever-increasing miniaturization of components in modern technologies makes characterization even more difficult. Sensitive nodes are now very close to each other, allowing the charge generated by a particle to affect more than a single node and even creating other failure mechanisms such as parasitic bipolar transistors between two adjacent devices, which can lead to catastrophic failures [35].

The variety of devices in a circuit, their complexity and miniaturization is responsible for the appearance of different effects. The usual classification of these Single Event Effects divides them between those that cause permanent damage to the device, and those that cause temporary damage, that disappears after certain time or after a reset or reconfiguration.

Destructive events

Permanent damage in microelectronic components can be triggered through different mechanisms and have different consequences to the device. Although in practice some of them are difficult to tell apart by their effects, the causes are well known. Among the most important destructive events caused we find Single Event Latch-up, Single Event Snapback, Single Event Burnout, Single Event Gate Rupture, and Single Event Hard Errors.

- Single Event Latch-up (SEL) is an unwanted high current state that mainly affects bulk CMOS devices. Latch-up is self-sustained, meaning that the parasitic structure cannot be undone unless a power cycle is forced. If the power supply is not cut fast enough, the high current flowing from source to ground through the low resistance path may destroy the device by thermal effect. SEL can be triggered when an ion track crosses the junction between the N-well and the P-substrate, creating a current path through the substrate to the ground. Fig. 2.5 shows the latch-up parasitic structure in a CMOS device [36].

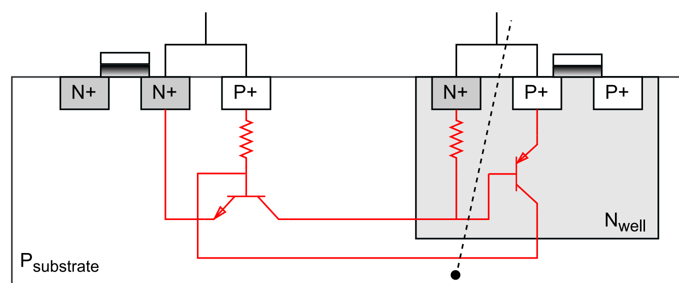


Figure 2.5: Latch-up path between the PMOS and NMOS transistors in the CMOS technology.

- Single Event Snapback (SESB) is a parasitic structure created in PMOS and NMOS transistors when an ion deposits charge in the p-n or n-p junction. This creates a parasitic bipolar transistor between the drain and source of the MOS transistor, that only disappears when the input polarity of the device is changed. Fig. 2.6 illustrates this phenomenon. SESB is not immediately destructive, but leads to performance degradation over time [37].

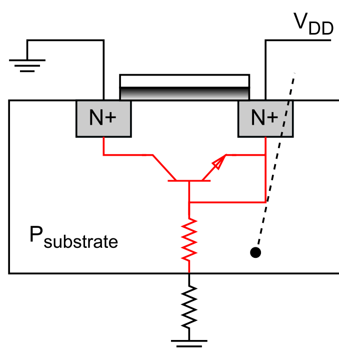


Figure 2.6: Single Event Snapback effect in a MOS transistor.

- Single Event Burnout (SEB) occurs in power MOSFETs due to the creation of a parasitic bipolar transistor, typically in the n-channel. This condition induces a regenerative feedback that causes the MOSFET to enter in avalanche state, creating a high current that ultimately leads to the destruction of the component. SEB is not a common issue in ASICs and FPGAs, but it can affect their power stages. Fig. 2.7 presents the Single Event Burnout parasitic path [38].

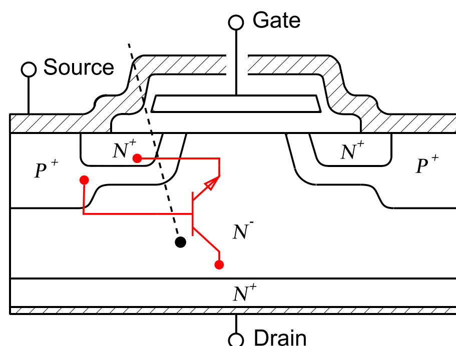


Figure 2.7: Parasitic structure of Single Event Burnout in MOSFET.

- Single Event Gate Rupture (SEGR) and Single Event Dielectric Rupture (SEDR) are triggered by incident ions in biased transistors. The deposited charge exceeds the breakdown field limit and ruptures the gate oxide or the dielectric layer. The induced current happens so fast that protecting against SEGR and SEDR is not possible, it is always destructive. Power MOSFETs are most susceptible to this kind of effects, but they can also be observed in linear integrated circuits and as stuck bits in digital devices. Fig. 2.8 shows the Gate Rupture mechanism in a MOSFET transistor [38].

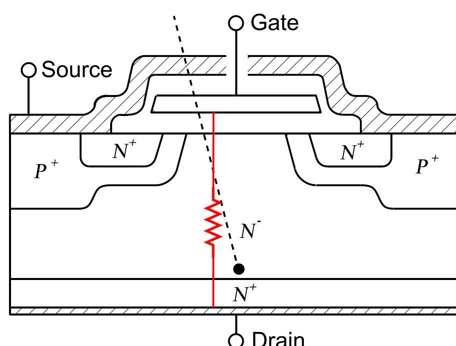


Figure 2.8: Single Event Gate Rupture effect in a power MOSFET structure.

- Single Event Hard Errors (SEHE) are a type of permanent errors found in memories and digital devices. Hard Errors appear as bits whose state cannot be changed anymore, known as stuck bits. These errors might recover given enough time. This is a topic under ongoing research, as the main mechanisms behind SEHEs are still uncertain [39].

Non-destructive events

Radiation particles can also inflict non-permanent damage in the electronic devices, both analog and digital. Nevertheless, transitory damage may produce catastrophic failures in a system, making the study of non-destructive events a necessity and the main topic of this Thesis. In contrast to hard errors, non-destructive events are also known as soft errors. Among the most important non-destructive events we find Single Event Transients, Single Event Upsets, and Single Event Functional Interrupts.

- Single Event Transients (SET) are temporary voltage spikes in a node of an integrated circuit. SETs are caused by single ionizing particle traversing the semiconductor near a sensitive junction. In analog systems, SETs introduce direct distortion in the signal, whereas in digital systems, SETs can affect the output of combinational logic gates and even be captured by memory elements [40] [41]. The capture of Single Event Transients by registers and other storage elements is one of the main mechanisms behind Single Event Upsets (SEUs) and Single Event Functional Interrupts (SEFI), which will be discussed later. A Single Event Transient being captured at the same time by more than one memory element would create multiple faults in different parts of the circuit, which will also be discussed later.

  The voltage spikes generated by Single Event Transients suffer different alterations as they propagate through the combinational logic and paths of the circuit. There are four phenomena related to SETs in digital circuits that may affect the chances of them being captured in a register, resulting in an SEU or a SEFI:

  - Logic masking: the combinational logic gates prevent the SET from propagating due to the logic values of the signals and the logical function implemented.
  - Latch-window masking: logical values at the input of flip-flops are stored only at the active edges of the clock signal. Due to that, SETs have a small time window to be latched in the register, known as window of vulnerability. Higher clock frequencies make the acquisition of erroneous data due to SETs more probable.
  - Electric masking: charge collected in the logic gates of combinational logic, as well as the resistance of the paths, may attenuate the voltage amplitude of the SET, eventually reducing it below the necessary value to change the state of a register.
  - Pulse quenching: the width of the voltage pulse is reduced as the result of charge collection mechanisms in logic gates. Charge deposited by an ionized particle may be shared between adjacent transistors in the same data path, causing them to change states at the same time. When the transient that affected the first transistor reaches the second, this second transistor is

set back to its previous state, thus reducing the amount of time it shows a faulty behaviour, "quenching" the pulse at the output. This effect increases as technology scales the size of the transistors and it was not detected in technologies larger than 130-nm.

  - Pulse broadening: Propagation-Induced Pulse Broadening (PIPB) is a temporal stretching of the voltage pulse induced by a SET. PIPB is caused by the hysteretic effect of the internal capacitance of a logic gate in the charging or discharging processes. A gate held at the same state for a long time may experience temporary changes in the threshold voltage of its transistors, making it faster (or slower) to react to SET voltage spikes. This produces pulse broadening, a negative effect, or pulse narrowing, which is a masking effect.

- Single Event Upset (SEU) are state changes in storage elements, such as flip-flops, latches or SRAM cells. Single Event Upsets can be generated by direct ionization of the storage element or by capturing the voltage change that occurs during a Single Event Transient. The effects of SEUs range from becoming completely silent, if the bit-flip affects an unused storage element or the wrong value is overwritten before being used, to creating errors in data words or control logic [41].

  Single Event Upsets usually affects a single memory element, but due to the increasing miniaturization of devices, the charge deposited by a single particle may affect multiple adjacent storage elements [42]. Multiple bit-flips can be triggered by different events and their effects may also differ:

  - **Multiple Cell Upset (MCU):** a single particle strike produces state changes in two different, physically adjacent, logic cells.
  - **Multiple Bit Upset (MBU):** this is a particular case of a Multiple Cell Upset that affects two memory elements in the same data word. MBUs cannot be corrected by simple Error Correction Codes.
  - **Common-Mode Failures (CMF):** this kind of multiple bit-flip is a specific case of a Multiple Cell Upset, in which the affected cells are part of two of the redundant copies used in Radiation Hardening By Design methods, which usually renders the hardening inoperative.

  Mitigating or reducing the occurrence of this kind of multiple bit-flips errors is still a challenge under study, because most of them are difficult to detect and correct using traditional hardening approaches.

- Single Event Functional Interruption (SEFI) are soft errors that may affect Finite State Machines or control logic among others in a device, causing malfunctions, resets or hangings. Most devices in modern systems require control logic, from memories of all kinds to microprocessors, ASICs, FPGA implementations or

mixed-signal electronics. A device affected by a SEFI usually requires a software reset or a power cycle to be recovered.

Fig. 2.9 shows the effect of different Single Event Upsets on the data words of an SRAM memory.

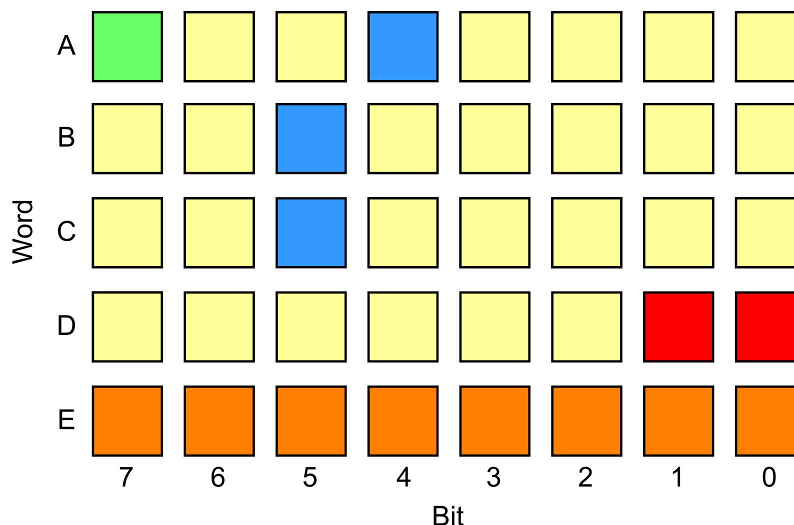


Figure 2.9: Examples of Single Event Upsets in an SRAM memory. The green cell represents a Single Bit Upset, the blue cells represent a Multiple Cell Upset, the red cells are affected by a Multiple Bit Upset and the orange cells represent the wrong values stored in the memory after a SEFI. [42]

Table 2.1 summarizes the effects caused by radiation in electronic devices.

Table 2.1: RADIATION EFFECTS ON ELECTRONICS

Effect	Caused by	Mechanism	Damage
TID	Photons, electrons	Hole trapping in oxide interfaces	Cummulative
DD	Protons, electrons	Distortions in the crystalline matrix due to collisions	Cummulative
SEL **SESB** **SEB** **SEGR** **SEHE**	Ions, electrons, protons, neutrons, muons	Ionization tracks creating shortcuts between different parts of transistors	Single Event, destructive
SET	Ions, electrons, protons, neutrons, muons	Voltage spike in a node of the circuit	Single Event, non-destructive
SEU		State change in a single memory element	
MBU/MCU/CMF		State changes in multiple memory elements caused by a single particle	
SEFI		Malfunctioning in control logic that affects its functionality	

2.3. Space Industry

The interest in space-related developments began after World War II, when the two main geopolitical blocks, countries supporting the Union of Soviet Socialist Republics (USSR) and countries allied with the United States of America (USA), began the research of propulsion systems capable of reaching the necessary velocity to escape the atmosphere. Although disguised as a scientific race towards space exploration, the real objectives were oriented to the deployment of nuclear weapons with enough range to reach the other block in case of war. The successful launch of the first man-made satellite, the Sputnik I, by the Soviet Union in 1957 led to the creation of the National Aeronautics and Space Administration (NASA) by the government of the USA, as well as two other military departments related to space research. Their efforts resulted in the launching of the Explorer I, the satellite used by James Van Allen to study the radiation belts around the Earth, in 1958 [43].

The USSR and the USA continued funding their space programs, the Soyuz and the Apollo respectively, with the intention of performing manned landings in the Moon, until this was achieved in 1969 by the American crew of the Apollo 11. The political tension between the two countries relaxed in the next decade and their space agencies even started cooperating in the Apollo-Soyuz joint space program.

During the first fifty years of space development, the main driving forces were the military, scientific and telecommunication applications, not only, but mainly funded and directed by governments and public institutions to fulfill their own objectives. Some private companies collaborated in the development of satellite-based television broadcasting, satellite telephones or Global Navigation Satellite Systems (GNSS), but the missions were planned and coordinated by public agencies [44].

The usage of public funds requires stringent justification and, although the revenue of the missions does not need to be necessarily economic, a high success rate is preferred, to avoid bad press reputation, that may have political consequences. This led to the creation of design and development strategies to avoid any kind of failure in the mission.

The so-called Traditional Space is characterized for the long development time windows, that usually span years, to minimize human error introduced accidentally in the designs. Exhaustive testing using established protocols in all parts of the system ensures that the mission will not suffer any catastrophic failure during its expected lifetime. Designing with so high safety margins, traditional space missions usually last longer than their expected service time, which is usually more than a decade [45].

In the past 20 years, new relevant actors other than public institutions have entered the space market, shifting the design paradigm towards a faster and not so reliable functioning in search for lower production costs and economic profit. This "New Space" trend tends to the planning of shorter missions, with also shorter development times and not so stringent requirements; failures are not only expected, but welcome, as long as a lesson is learnt

and not repeated again. Contrary to popular belief, this new trend has not been caused by the lack of public funding in research and development, in fact, the public budgets have increased all over the world [46]. Various are the reasons behind this shift in the industry, from the more accessible information worldwide, to affordable and powerful electronic devices and the enormous initial investments in research and development made by governments throughout the years. Of course, the recent investing efforts of individuals and capital risk companies are also responsible and will play an important role in the coming years [47].

The first space missions used the commercial components that were available in the incipient transistor-based electronics market. Soon after, the first reports of the effects of radiation on space electronics were published and the industry focused in protecting the already existing parts and devices against the effects that were identified at that time. This strategy is nowadays called Radiation Hardening By Process (RHBP). Modifying the fabrication process to tackle the effects of radiation renders good mitigation results, but is an expensive decision: scale economies do not apply when the number of fabricated devices is small. Moreover, the research on RHBP cannot keep up the pace with the rapidly evolving transistor technology and Radiation-hardened (Radhard) parts are always less efficient and with lower performances than their commercial counterparts. However, under the premise of avoiding any failures that is the base of Traditional Space, the investment in RHBP makes sense.

The pursuit of lower development costs and higher performances in the space industry has brought back the interest in using commercial devices as part of space systems, mainly for non-critical subsystems and payloads. The New Space industry is leaning towards the usage of Commercial Off-The-Shelf (COTS) components and radiation-hardening strategies based in a carefully planned design able to detect and correct errors before failures occur (Radiation hardening By Design (RHBD)). The selected components are usually military or automotive graded, which have higher quality and durability standards, but still affordable costs and market lead times. This approach allows for the space systems to have state-of-the-art technology, becoming more than a sensor or a communication link, being able to execute complex calculations and preprocess some data before sending it back to Earth-based stations. Nevertheless, a minimum qualification of the behaviour of each COTS components against radiation is needed to minimize the risk of fatal failures, such as Single Event Latch-up. Testing the limits of components beyond the recommendations of the vendor (up-screening) is another usual procedure in COTS components. Since an exhaustive qualification and up-screening process is expensive, the budget and time limitations of the project determine the experiments performed.

These New Space approaches have one major downside regarding the environmental impact of massive production of short-lived systems. The number of small satellites orbiting the Earth is planned to be greatly increased in the next decade, providing mobile communications and other services in remote areas where landlines are difficult to implement. These satellites will be deployed in mega-constellations, nets of

intercommunicated nodes able to provide system redundancy if any of them is affected by faults and reaches its end-of-life. Satellites becoming unresponsive or intentionally shut down are at risk of remaining in orbit for a long time, as space debris, compromising the correct functioning of other orbiting systems [48]. Modern satellites are required to follow strict protocols that ensure safe re-entry in the atmosphere before becoming space debris. Disposal maneuvers are not only required for satellites in protected LEO and GEO orbits, but also for launch vehicles, whose landing must be confined to a certain region [49] [50].

2.4. Dependability and qualification

Due to the difficulty of repairing and maintaining space systems and the high costs associated to their development, missions must be carefully planned before launch to ensure the objectives of the mission are met. This process includes a characterization of the mission and the environment that will surround it, as well as performing several tests in the parts and systems that will be used. The degree of confidence placed in the success of the mission can be examined through different prisms and objectives, and measured in different ways depending on the application.

In this section we will review some of the terminology related to dependability of the system and how to predict the performance of the system in space by previous experimentation.

2.4.A. Dependability

Dependability can be defined as the extent to which the fulfilment of a device can be justifiably trusted. Dependability is a compound measurement of the reliability, maintainability and availability of the system and should be taken into consideration along with safety. Other aspects related to dependability, such as prevention, removal, tolerance and forecasting of faults, shall be taken into consideration [51].

- Reliability can be defined as the ability of a system to carry out a function under given circumstances during a defined time span. The selected time interval for space missions is usually the length of the mission, which can vary between months and decades depending on the objectives.
- Maintainability describes the probability that a maintenance action can be performed on a system during a given time interval and using predefined procedures and resources.
- Availability is the capability of a system to be in a state to carry out a required function under given conditions in a specified time frame, assuming the necessary external inputs are provided.

The term safety can be described as the state of a system in which an acceptable level of risk is not exceeded. Risk in space missions is related to injuries in manned flights, damage in launch facilities, damage to human-machine interfaces, damage to the main functions of the system itself or environmental damage.

The methods to ensure system dependability can be classified in four major groups: prevention, removal, tolerance and forecasting [52].

- Fault prevention aims at preventing the introduction of errors in the final system. Strict procedures are required during the definition, design and development of a device, be it software or hardware. Design and coding standards help reducing human error and identifying problems.
- Fault removal is a way to detect faults and remove them from the system. Verification and validation through emulation or simulation are the most common approaches to fault removal.
- Fault tolerance is conceived as a means to avoid that faults lead to system failure, by detection and correction of faults occurring while the system is functioning. This way, fault tolerance techniques, such as exception handling or data correction, prevent the faults to propagate and keep the system working properly.
- Fault forecasting is aimed at estimating the fault incidence of the system by analyzing the results of previous data of the device working under similar circumstances. Fault forecasting can be performed as a qualitative analysis, with methods such as Failure Mode and Effect Analysis (FMEA), or quantitative analysis, through statistical models.

Fault forecasting: measuring dependability

Prior to the launch of the system to the real working conditions, several tests under similar circumstances must be carried out to the different components of the system and the system as a whole. The objective of these experiments is twofold: first, to identify faults introduced in the manufacturing process, and second, to predict the fault sensitivity of the system once it has been launched.

The failure distribution during the life of any component can be described as in Fig. 2.10. At the beginning of the life of the component, the probability of an occurring failure is high due to errors in the manufacturing or assembly process; this phenomenon is known as "infant mortality". The wearing out of the component near the end of its life leads to higher error rates until it stops functioning correctly. The span between these phases is the useful life of the device, in which the error rate is almost constant. The idea behind tests performed previous to launch is to situate all the devices in the "useful life" phase, avoiding unexpected faults [53].

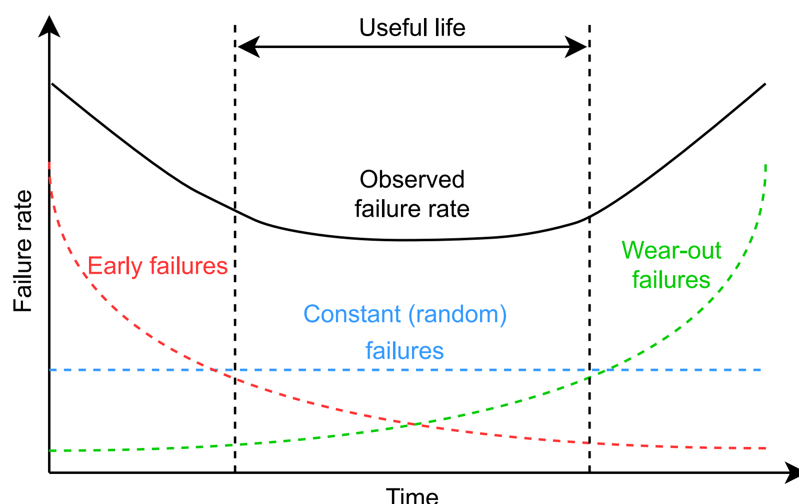


Figure 2.10: The Bathtub curve describes the failure rates of any device during its cycle of life.

The behaviour of the system can be predicted using the results from previous experiments. The experimental results can be expressed using various figures of merit, that allow us to characterize the behaviour of the system, as well as comparing different systems.

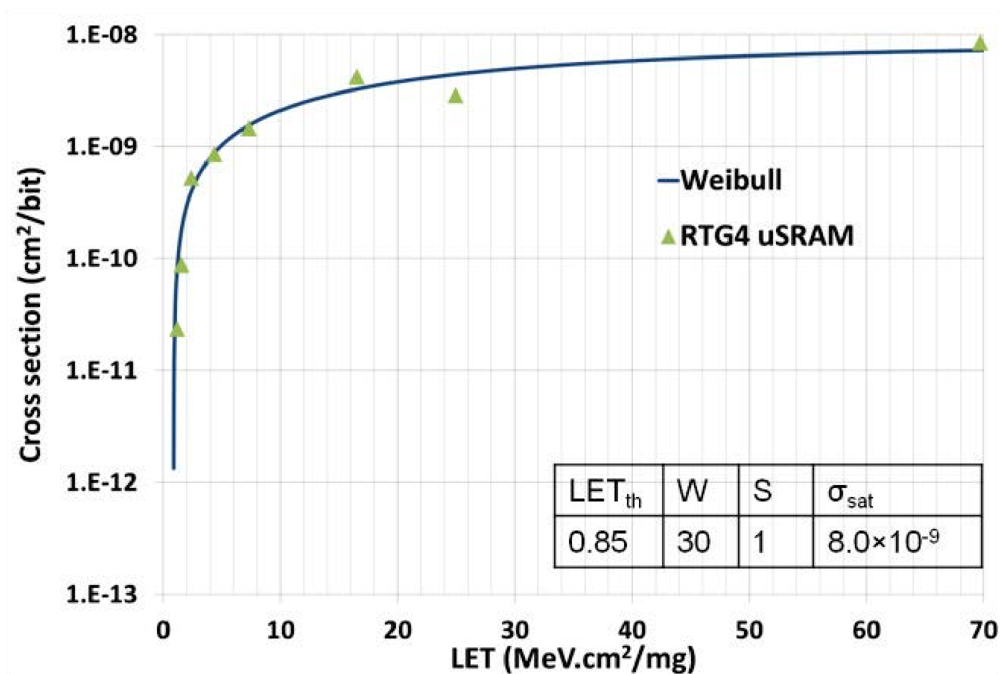
- **Cross Section**

  To determine the sensitivity of a component to faults induced by ionized particles, the most usual measure is the cross section (σ). The cross section is calculated as the number of faults detected during the irradiation campaign divided by the fluence (Φ) of the particle beam, where the fluence is the number of particles that traverse a unit of area. Eq. 2.3 describes the procedure to calculate the cross section.

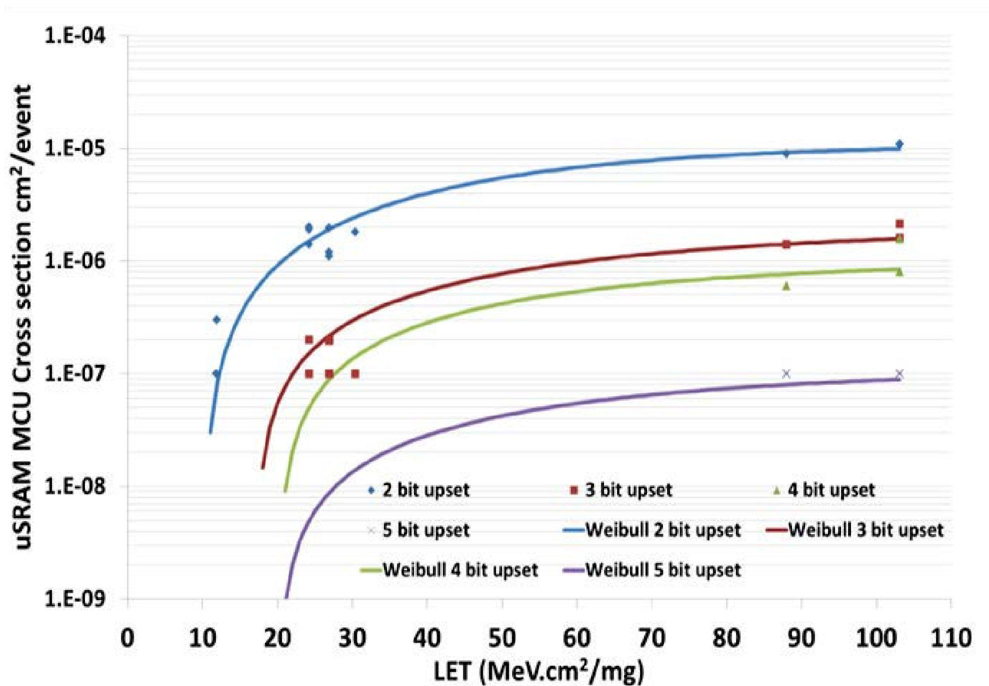
$$\sigma = \frac{\text{Number of faults}}{\Phi[particles/cm^2]}[cm^2] \tag{2.3}$$

  The Cross section is usually referred to any kind of Single Event Effect produced in a device by ionizing radiation depending on the behaviour that needs to be studied. So, we could calculate the total SEE Cross section or a more specific Cross section to measure the sensitivity to a particular effect.

  When performing irradiation experiments, the Cross section is usually represented as a function of the Linear Energy Transfer (LET) of the particles in the beam. The resulting graph follows a Weibull distribution as the one shown in Fig. 2.11. This curve serves the purpose of characterizing the behaviour of a certain device and can be used later to predict its behaviour under different environmental conditions [54].



(a)



(b)

Figure 2.11: Cross section vs. LET curves in an SRAM memory: (a) refers to total Single Event Upsets found, while (b) shows curves for different types of Multiple Bit Upsets in the memory [54].

A measurement analogous to the cross section can also be obtained using fault injection experiments instead of irradiation campaigns. This figure of merit is called

error rate in this Thesis and can be calculated as the number of errors divided by the number of faults injected in the design, as shown in Eq. 2.4. The error rate cannot be directly compared to the cross section, but there is a certain correlation between them that makes injection experiments a valuable tool to test designs before irradiation campaigns.

$$\text{Error rate} = \frac{\text{Number of errors}}{\text{Number of injected faults}} \tag{2.4}$$

- **Failure rate**
  When analyzing the sensitivity of a system to any kind of failure, not only to those caused by ionizing particles, the most common approach is to evaluate the level of reliability using the probability of failure, also known as the Failure Rate (λ). The Failure Rate can be calculated as the number of failures per hour found in a system under operating conditions. Eq. 2.5 shows how to compute this number. The variables present in this equation are f, the number of faults, n, which represents the number of samples tested, and t, the cumulative experiment time expressed in hours [55].

$$\lambda = \frac{f}{n \times t}[failures/hour] \tag{2.5}$$

  The Failure Rate can be used to establish the reliability level of a system in different ways, depending on whether the system is repairable or not.

  - **Failure In Time (FIT)**
    Failures In Time is a standard measurement derived from the Failure Rate. It is defined as the estimated number of failures of a system in a billion hours of functioning, as seen in Fig. 2.6.

$$\text{FIT} = \lambda_{FIT} = \lambda \times 10^9[failures] \tag{2.6}$$

  - **Mean Time To Failure (MTTF)**
    Mean Time To Failure is the standard measurement used in the industry for non-repairable systems. MTTF is also derived from the Failure Rate and it offers a view of how often failures are expected to happen. Eq. 2.7 shows how to compute the MTTF.

$$\text{MTTF}_{hours} = \frac{1}{\lambda} \tag{2.7}$$

  - **Mean Time Between Failures (MTBF)**
    As for repairable devices, the most common figure of merit to calculate the level of reliability is the Mean Time Between Failures. The MTBF takes into account the Mean Time To Failure as well as the average time needed to perform a reparation, the Mean Time To Repair (MTTR), as in Eq. 2.8

$$\text{MTBF} = \text{MTTF} + \text{MTTR} \tag{2.8}$$

2.4.B. Radiation Hardness Assurance: Qualification

The process of selecting components for a space mission, not specifically electronic components, but all kinds of devices, starts with a description of the design requirements of the mission, usually dependent on its objectives. Design requirements must take into account physical characteristics as size, weight and power and also more technical aspects like data transfer speeds, availability or downtime.

Regarding the radiation behaviour of the system as a whole and of each individual component, the first step is to characterize the environment that the system will be subject to. With this information, parts can be selected that fulfill both the technical requirements and the radiation hardness requirements related to the environment and the length of the mission. This is true for many parts, especially radhard ones, where radiation performance has been assessed by vendors or other researchers. However, sometimes it is necessary to test components under specific conditions to qualify them, as is the case of COTS parts. For COTS, the manufacturing variability is also a key factor. Lot-to-lot differences must be taken into account to avoid different behaviour between the tested parts and the flight models [29].

In this section we will briefly cover the process of selecting parts, focusing particularly on radiation testing methods, some of them will be discussed later on in this Thesis.

Environment characterization

When planning a mission that may be affected by radiation, it is important to perform a thorough research on the environmental conditions of the region which the system will be orbiting or will traverse. As different environments are populated by different kinds of particles and these particles have different effects on electronics, the characterization must be performed as the first step of the study. For this task, software such as SPENVIS (SPace ENVironment Information System) [56] and OMERE [57] are free-to-use and provide detailed information about the environment, including expected fluences of particles and TID and DD doses over the duration of the mission.

Traceability

Once the environmental and technical conditions have been determined, the adequate parts and components can be selected from the different vendors' catalogs. Traditional space missions usually prefer radhard parts, but for many non-critical functions, COTS can be selected. Special technical requirements enforce the usage of commercial parts, due to their higher performances. In both cases, the next step would be to procure enough commercial parts, ensuring the homogeneity of the lot, to carry out qualification experiments.

There are two aspects to consider regarding the manufacturing of COTS: complexity

and mass fabrication.

COTS use standard manufacturing processes, which are usually composed of a few hundred very complex steps involving physical, optical and chemical transformations of the silicon wafer in which the circuit will be created. Small variations in the optical parameters of the lithography process, the surface of the wafer or temperature adjustment may result in the introduction of non-uniformities in the circuit, which lead to variation in the response and, in extreme cases, to defects or failures [45].

Mass-production, however, should be able to correct this kind of defects, especially for automotive and space grade components, due to thorough quality tests. Moreover, with a high number of parts working at the same time all over the world, the probability of undetected design errors tends to zero. These two factors play in favor of fault prevention and fault removal in COTS.

Nevertheless, mass production has a major drawback for space applications: in order to reduce costs, chips vendors usually spread their production and assembly lines over different foundries. Manufacturing equipment is usually different in those fabrication plants, and even if they are the same, hardware, software and procedures may differ too. This produces the so-called fab-to-fab variation, which can be even greater than lot-to-lot variation and must be meticulously examined in order for the qualification and up-screening results to be valid.

Traceability refers to the ability of recovering the historic of modifications that a piece of hardware suffered back to its origin [51]. Ensuring traceability in COTS is important because of fab-to-fab variations. In order for the testing results to be extrapolated to every purchased part, the components must be as similar as possible. Two COTS with different origins are not guaranteed to behave in the same way when exposed to radiation, whereas their response should be similar enough if their traces are the same [58].

Published qualification and up-screening results are as important as the trace of the tested devices, so, even if there are already available qualification data of a commercial part, testing must be performed again if their origins cannot be trusted or are different from the origin of the available parts in the moment of selecting a component for a mission. Efforts towards predicting the behaviour of components with different origins by analyzing previously published results have been made with good results [59], but acquiring enough and detailed data to perform an statistically relevant analysis is difficult for most components.

Testing

The last step of the part selection process is testing the devices for the type of faults expected depending on the environmental conditions required by the mission. The behaviour of the devices is usually tested for three parameters: its sensitivity to Total Ionizing Dose, to Displacement Damage and to different types of Single Event Effects.

Additionally, modern irradiation facilities allow for small systems to be tested as a whole under operating conditions, which is known as System Level testing.

It is worth noting that for all these testing campaigns, the fluxes of particles are much higher than the real fluxes that will affect the system once in real operating conditions. The experiments are, thus, accelerated versions of the real environments, made this way to allow for faster acquisition of samples.

In this section we will briefly review the most relevant aspects of the TID and DD test campaigns, while delving deeper in the Single Event Effects testing, that is the main experimental method used in this Thesis.

- **Total Ionizing Dose testing**
  Recommended guidelines for TID testing are provided by the European Space Components Coordination (ESCC) in their ESCC22900 standard [60]. According to this standard, these experiments can be carried out with gamma rays (using a Co-60 source) or protons, although alternative sources like X-rays or electrons can also be considered with the proper data treatment. Gamma rays (photons) are not a problem in space, due to their very low concentration, but they have become the most used testing source, because they are a conservative approach to mimic the effects of protons and electrons in terms of TID [61].

  TID experiments consist in irradiating the devices with the preferred source of particles during a certain period of time under the worst-case scenario, until the total expected dose absorbed by the device during the mission is reached. Then, the characteristics of the device are measured to annotate the final degradation suffered.

  Depending on the Radiation Hardness Assurance level required and the selected Dose Rate, the experiment can take several one-hour irradiation phases followed by recommended performance measures in the device. This is advisable in order to detect possible malfunctions at lower absorbed doses and being able to print a complete TID degradation versus Dose curve.

  After the total expected dose has been administered to the device, a 24 hour annealing period at room temperature is usually waited for another measurement of the performance. Ageing the irradiated device by increasing the room temperature by 100 ºC during 7 days is the final step of the experiment, after which, a final measurement is collected at room temperature.

- **Displacement Damage testing**
  The ESCC also elaborated a document providing guidelines to perform Displacement Damage tests in electronic devices, the ESCC22500 standard [62]. Following the recommendations of this document, DD testing can be carried out either with a proton or a neutron beam. Since protons are charged particles, also contribute to create TID effects on the device; the total dose accumulated in the device during a DD experiment must be calculated and presented along the DD results.

In the case of using proton accelerators, care must be taken not to degrade the flux of particles to lower the energy of the incident protons. The flux must be constant and uniform in the region of interest where the sensitive volume, the component to be tested, is located. Adjusting the shape of the beam and lowering the energy of the particles that reach the system is usually achieved by placing collimators and degraders in between the source and the device. These components are metallic blocks that can be placed automatically and remotely at any point during the experiment, which makes it a fast and convenient way of adjusting the energy. However, the particles can be deviated by the degraders, risking the uniformity of the beam after them, which is advised not to have variations larger than 10%. Other adjustments of the beam energy, although time expensive, are preferred to avoid this effect.

For this same reason, while testing with low energy protons, it is recommended to remove the plastic or metallic lid covering the chip so that particles with enough energy are able to reach the sensitive volume.

When designing Displacement Damage experimental campaigns, the main objective is to irradiate the part up to a certain fluence level, that is the approximate number of particles traversing the component. The fluence level is given by the mission requirements. Thus, the testing steps would be to measure the characteristics of the tested devices previous to the experiment, irradiate the component until the desired fluence is achieved and measure the deterioration in the performance of the device due to Displacement Damage. Measuring at other fluence levels is also recommended, namely at half and double the nominal fluence level.

- **Single Event Effects testing**

  Single Event Effect testing campaigns are performed in order to characterize the sensitivity of a circuit against the plethora of destructive and non-destructive events that can affect a device in the space environment. The experiments must be designed according to the type of trait that is wanted to be detected or measured, with the ultimate goal being to calculate the sensitivity of the device to a certain type of event.

  In this section we will review the whole Single Event Effect testing procedure, from the selection of the beam parameters to the analysis of the experimental data.

  - **Guidelines and methods**

    Testing for Single Event Effects is very application-specific, so the available documentation regarding SEEs cannot cover all the possible cases, they are mere guidelines to help radiation engineers in the most common experiments.

    ESCC25100 is the most general document available, providing methods and recommendations to test with all types of particle beams and also test for different types of Single Event Effects [63]. Other documentation, as the ASTMF1192 [64] or the JESD57 [65] are focused on the methodology to irradiate parts using heavy ions. The JESD234 [66] advises about proton testing to measure SEE, while

JESD89 [67] describes methods to irradiate using neutron beams and extrapolate the results to terrestrial effects.

Other approaches recommend protocols not for types of particles, but for the effect to be investigated. This is the case of the MIL-STD-750 [68]. The SEE testing methods explained in this document refer specifically to Single Event Burnout and Single Event Gate Rupture in power MOSFETs.

– **Designing the experiment**

When performing SEE testing, the first step is deciding the type of effect that we want to detect. This is specially relevant in complex devices, which usually contain different types of submodules subject to faults. For example, analog devices are sensitive to Single Event Transients, power stages may suffer destructive SEEs, memory elements are affected by Single Event Upsets, and Single Event Functional Interrupts usually affect Finite State Machines. Designing the adequate instrumentation to detect and register the selected events is key for the irradiation campaign.

Protecting the device against destructive effects, as well as against other undesired SEEs is also important in order to bring the irradiation campaign to a successful end.

– **Choosing the facility**

The requirements of the mission determine the effects that the device will be affected by, as well as the energy spectrum present in the region. In order to perform the test under the most comparable circumstances as the real conditions, it is important to select the adequate particle beam.

Protons and heavy ions are the two recommended beams to test for Single Event Effects, however, depending on the characteristics of the experiment and the availability of the test facilities, other methods may be applicable.

When choosing the right beam for our experiment, several beam parameters must be observed, such as the energy and flux ranges, the beam profile and purity and the energy spectra of the particles. There are other physical parameters that can be adjusted in the facility, such as the need of vacuum chambers for heavy ion experiments, modifying the size and shape of the beam with degraders and collimators or adjusting the distance and tilt of the device under test with respect to the beamline.

Paying attention to the data and power connections that communicate the irradiation chamber with the control room in the test facility is also important to carry out the experiment satisfactorily. Although the amount and variety of connections in the facilities is usually enough to cover all the necessities of the experiment, the connections must be carefully planned in the design phase, trying to reduce the complexity of the cabling and the presence of measuring equipment that may be subject to radiation.

– **Preparation**

Heavy ion penetration is usually low and chip packaging may reduce the energy of the particles reaching the device and even stop them. To ensure that a significant amount of particles affects the device under test, removing the packaging may be needed, depending on the chosen particle, energy and encapsulation. Delidding is not necessary for proton testing.

Fans and heat sinks in the path of the particles must be removed in any case, because they can alter the homogeneity of the beam profile as well as reducing the energy and producing secondary particles. This may cause heat dissipation problems for some devices and experiments carried out in vacuum may become totally impracticable.

Mechanical and chemical procedures can be used to decapsulate and remove package lids. These operations risk the integrity and functionality of the device, as there is a significant probability of damaging several components, so procuring extra parts is mandatory to avoid delays in the experiment. Testing the parts after the delidding process is necessary to ensure that there are no errors that can later appear during the experiment and distort the measurements.

– **Test campaign**
  During the radiation campaign it is advisable to monitor the development of the experiment as closely as possible making fine adjustments in the fluence or energy of the beam in order to reach an acceptable SEE rate. Having spare parts and materials, extra cabling and alternative designs to test is also recommended to fix problems that may arise during the experiment.

– **Other sources**
  Although not recommended in the standards for space applications, other methods to provoke SEEs in the device under test may be used. The use of these alternative methods is justified by the kind of effect that is under analysis or to isolate the effect of particles hitting a sensitive small area of the chip. These sources can also be used to validate proton and heavy ion results or to investigate the sensitivity of a device prior to irradiation.

  * **Neutrons**
    Neutrons are not a relevant particle in space environments, but are the main source of SEE in terrestrial installations (communications, automotive, medical) and aviation equipment. For this reason, the study of the effects of neutron irradiation is interesting by itself, but using neutrons can also provide reliable information about SEUs, comparable to the results obtained in proton and heavy ion irradiation [69]. For SEUs and other SEU related SEEs, performing neutron irradiation may be justified if the fault mechanisms are not as important as the effects of the SEU on the system.

  * **Laser testing**
    Charge deposition directly in the chip can be done using laser pulses. There are two electron-hole pair generation mechanisms achieved by laser pulses:

Single Photon Absorption (SPA) and Two Photon Absorption (TPA). The SPA mechanism deposits charge in larger areas and can be used to carry out chip-level testing, whereas the TPA mechanism has higher precision and can be used to inject charge at cell-level. As in other testing approaches, numerous parameters of the laser such as the energy and frequency of the pulse or the spot size can be adjusted to emulate the charge deposition of a particle at different depths and positions of the silicon. The main interest of laser testing is mapping the sensitivity of a device, depositing charge in small areas over a regular grid in the chip, although random charge deposition can be performed as well [70].

  * **X-rays testing**
    Using very similar mechanisms to those of pulsed laser testing, pulsed microbeams of X-Rays can be employed to cause SEEs in electronic devices that can be correlated to the results of proton and heavy ion experiments [71].
  * **Fault injection testing**
    Fault injection testing is based on the instrumentation of a device to emulate SEUs or SETs in certain components of said device. This way, the effects of the injected fault and how it propagates through the device can be observed. Fault injection is a powerful and inexpensive technique to investigate the impact of SEEs in a hardware or software design in an exhaustive manner. Faults can be injected in a sequential or random way and at the desired execution moments, allowing for a very flexible and precise testing [72]. Tools and strategies for automatic error injection in digital circuits are available for multiple platforms and vendors [73], [74], [75] [76].

- **System Level testing**
  Although it minimizes the risk of failure during the mission, performing SEE tests in every component of a complex system is expensive in time and resources. A faster and less expensive test regime is performing the radiation experiments at system level. This approach allows for faster qualification of systems, which is in accordance to the New Space development trends, but is also challenging and should not replace component level testing of critical parts.

  System level testing is intended to verify the reliability and availability of the final application running in the system, instead of more simple applications tested in each individual component of the system. This way, the observability of the test is worse, since knowing where the error started and how it propagated is difficult in a complex system, but the test conditions are more similar to the real working conditions of the system in orbit. The outcome of the experiment is a pass or fail result, which may come at high costs if the whole system must be redesigned to mitigate the observed faults.

  System level radiation experiments pose some difficulties, since maintaining the homogeneity of the beam is complicated in systems with big physical areas and different shielding layers. Decapsulation and using vacuum chambers big enough for heavy ion testing may be difficult or even impossible, so using high-penetrating beams

or spallation chambers is recommended. A mix of high energy protons and neutrons is commonly used for this task. However, this does not allow for a proper research on the sensitivity to destructive SEE or TID of the device. Moreover, in case a single System Under Test is available, performing destructive or cumulative damage tests is not possible.

For these reasons, system level testing should be regarded as a complement for component level tests or as a final verification step for complex devices whose subcomponents have been already tested separately. Overall, system level testing should not completely replace component level testing, but testing the most critical components individually and performing a final test at system level could be a good trade-off in terms of time, money and reliability [77].

All the reviewed testing standards also propose ways to catalogue and publish the results of the testing campaigns, allowing other researchers to access the complete record of the irradiated component, including its traceability information and the irradiation performance under given conditions.

2.5. Hardening against radiation

Protecting electronic components and systems against the effects of radiation is a major concern in the aerospace industry. Great efforts are put into mitigating permanent and transient faults at different levels of the electronic components, trying to find optimal solutions at lower development and manufacturing costs.

In this section we will review the most common hardening methods that can be implemented in electronic systems to counter the different effects of radiation on the circuit. Since this is the main topic of this Thesis, we will place particular emphasis on the error mitigation techniques for Single Event Upsets based on design modification approaches.

2.5.A. TID hardening

Protecting devices from the degradation caused by Total Ionizing Dose mechanisms usually implies a combination of modifications in the manufacturing processes, known as Radiation Hardening By Process (RHBP), and special layout design techniques. The main goal of these two approaches is to reduce the amount of charge trapped in sensitive areas of transistors, that may ultimately worsen their performance over time.

Modifying certain parameters in the fabrication process may either increase electron trapping or decrease hole trapping, both of which reduce the amount of charge trapped, ultimately increasing the TID hardness. Reducing the temperature in the growth process is a good measure to avoid losing TID hardness, as an inverse relation between growth

temperature and hardness has been demonstrated. Using argon during the growth or Post Oxidation Annealing (POA) process does not degrade the TID hardness as nitrogen does, which is commonly employed in commercial technologies. Implanting nitrogen in silicon prior to oxidation improves the TID hardness due to proton interactions.

Charge trapped in gate or Shallow Trench Isolation (STI) oxides are a major concern for TID hardness. However, this effect is only significant for oxide thickness higher than 5nm, present mostly in technologies above 250nm. The scaling trend favors the TID hardness, as technologies below 250nm do not show important TID degradation due to this effect.

Accumulation of trapped holes in the STI oxide leads to an increasing leakage current between the oxide and p-doped regions. Even for small technologies, some transistor designs still have thick oxide layers that make them sensitive to TID. Modifying the layout of the transistor to physically separate the p-doped and oxide regions is the most obvious solution [41].

Enclosed Layout Transistor (ELT) or Ringed Transistor is a technique to avoid leakage current between the drain and source of the transistor [78]. The idea is to encircle the drain by the narrow layer of polysilicon gate and place the source around them, separating them completely, as seen in Fig. 2.12.

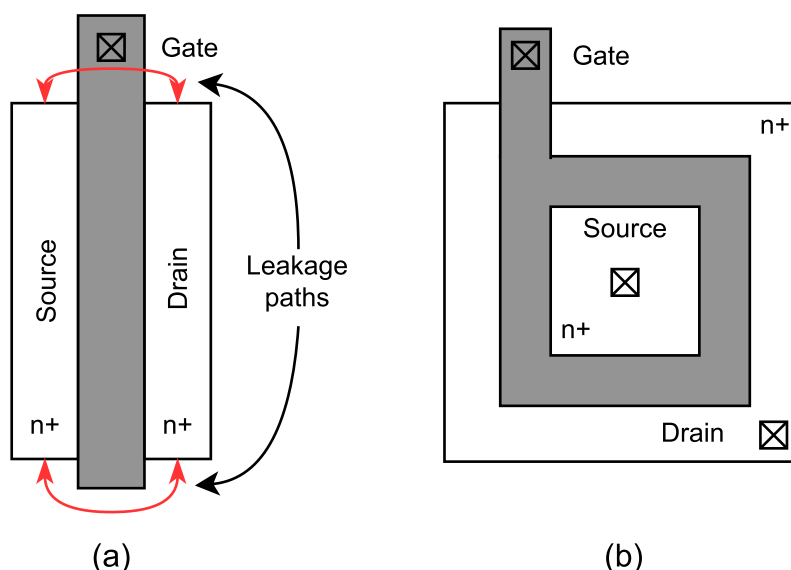


Figure 2.12: (a) Conventional NMOS transistor design, (b) Enclosed Layout Transistor design. [79]

Guard rings are large taps placed encircling a region of the silicon, as in Fig. 2.13. Placing guard rings around NMOS transistors can help reduce trapped charge in sensitive areas of the transistor, reducing its sensitivity to TID [80].

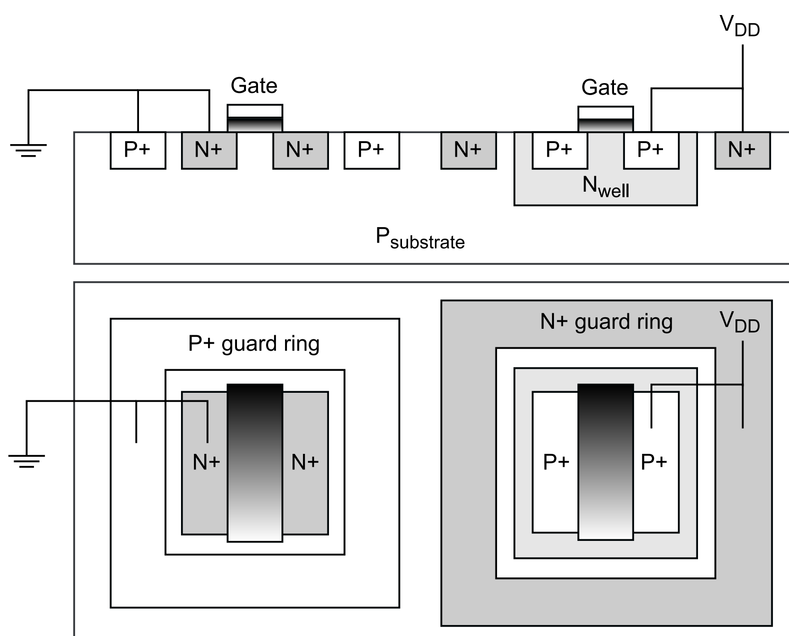


Figure 2.13: CMOS transistor protected with guard rings [80].

2.5.B. DD hardening

Current CMOS technologies used in ASICs and FPGAs are not sensitive to Displacement Damage because this effect mainly affects minority carrier lifetime, whereas CMOS are majority carrier devices. Other devices based on minority carrier transistors are sensitive to DD, but we will not cover hardening techniques against Displacement Damage in this Thesis, because it is not strictly related to the topic [81].

2.5.C. SEL hardening

As was the case of TID hardness, there are different approaches towards Single Event Latch-up hardness. They can be divided in Radiation Hardening By Process and Radiation Hardening By Design (RHBD) techniques. Among the RHBP mitigation techniques we can find the Silicon On Insulator (SOI), the Triple Well, and the Buried and Epitaxial layers techniques.

The Silicon on Insulator technique is an alternative manufacturing process in which the transistor is implemented on top of a silicon dioxide (SiO_2) or sapphire (Al_2O_3) insulating layer called Buried Oxide (BOX). This layer provides isolation between the wells and the substrate, blocking the appearance of parasitic currents that ultimately lead to SEL. The addition of the BOX layer leads, however, to a decrease in the performance against TID, since charge is easily trapped in the insulation oxide. Fig. 2.14 shows the difference between a conventional and a SOI transistor.

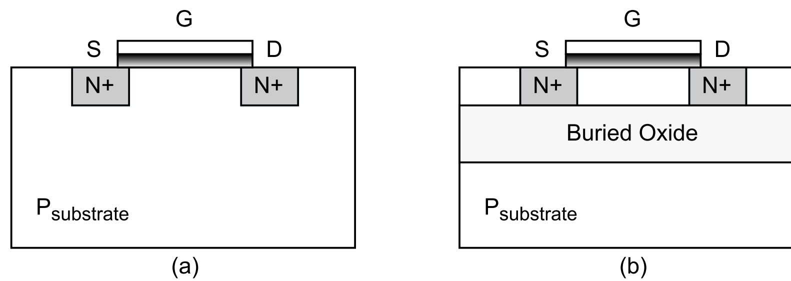


Figure 2.14: Conventional NMOS transistor (a) and NMOS transistor hardened using SOI (b).

SOI transistors also offer significant advantages in power consumption and performance, and allow for a higher device density when compared to conventional bulk CMOS manufacturing techniques. Additionally, SOI provides a certain SET and SEU hardness, that will be explained in the next section [82].

The Triple Well technique aims at eliminating the parasitic current paths between the wells and the substrate of the transistor. To do so, both PMOS and NMOS transistors in a CMOS device are isolated from the substrate by reversed biased junctions in the form of p and n wells [82]. Fig. 2.15 shows the structure of a Triple Well CMOS transistor compared to its conventional counterpart.

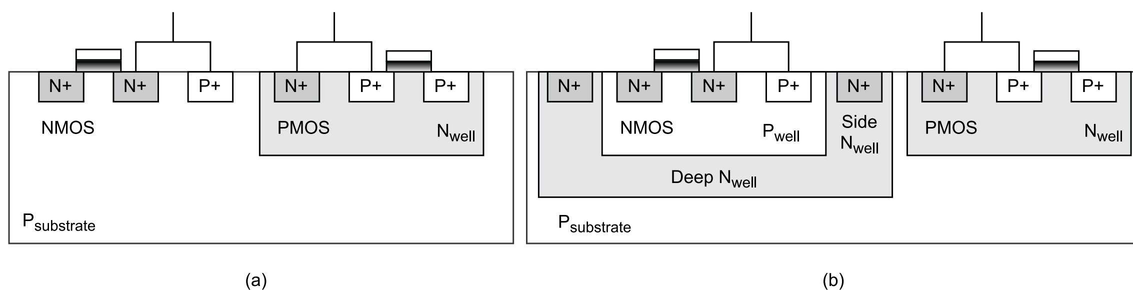


Figure 2.15: Conventional CMOS transistor (a) and CMOS transistor hardened using Triple Wells (b).

As was the case of the SOI technique, the addition of extra wells increases the sensitivity of the device to TID effects.

The Buried layer strategy consists in the addition of a highly doped layer inside a lightly doped substrate or well right beneath a sensitive node. The highly doped area is able to collect or repel the charge deposited by a particle, thus increasing the threshold charge needed to trigger a latch-up effect in that node [83]. Fig. 2.16 shows an NMOS transistor hardened using the Buried layer mitigation technique.

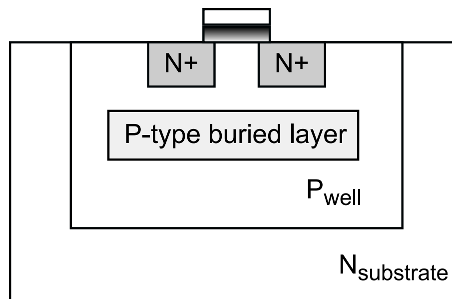


Figure 2.16: Buried layer in an NMOS transistor.

The last RHBD tecnique we will review is the growth of epitaxial layers on top of the substrate. Epitaxial layers are thin monocrystalline films that act as a high resistivity path between the highly doped substrate and the well, hindering the formation of parasitic circuits that cause latch-ups [83]. In Fig. 2.17 we show a CMOS transistor implemented on an epitaxial layer.

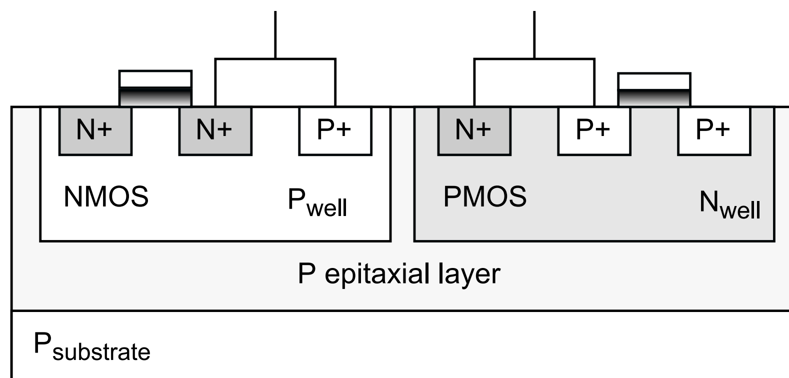


Figure 2.17: CMOS transistor hardened against SEL using a p epitaxial layer.

The most common RHBD technique to avoid latch-ups is the addition of current limiters. Latching Current Limiters (LCL) are active elements that protect the system from current overloads. This devices are based on the addition of a current sensing circuit that detects dangerous transients and a power MOSFET capable of cutting the power line of the system with small reaction times. A basic LCL configuration is described in Fig. 2.18. In this circuit, a small resistor is placed in the power line and the voltage it senses is amplified to drive the power MOSFET. A latching circuit is added to the system to avoid that the power is restored before the transient has been mitigated. Modifications can be implemented in the basic LCL to allow for current regulation during the transient, instead of completely shutting down the affected subsystems [84].

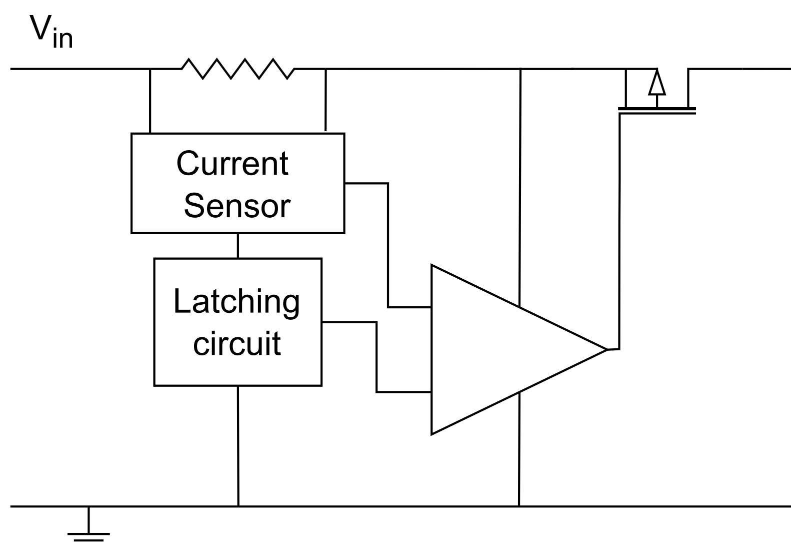


Figure 2.18: Basic configuration of a Latching Current Limiter.

In complex devices, LCLs may be applied at different hierarchical levels of the design, allowing a fine control over the subsystems that are shut down in case of latch-up. This also allows for a fine tuning of the current threshold that triggers the LCL, specially in devices where the processing load produces significant variations of the working current. Setting individual trigger thresholds for each component allows for a better protection of the whole system.

Updating the threshold values to account for the TID degradation of the components can be carried out in some reprogrammable systems with the addition of an ADC to the system to monitor the typical current usage of each component protected by an LCL [85].

2.5.D. SET hardening

Hardening devices against Single Event Transients can be approached in different manners. Some of the previously explained RHBP and layout techniques add a certain SET protection, as well as the described reduction of latch-up or TID sensitivity. Design modifications in analog and digital circuits can be performed to reduce the duration and intensity of the SET pulses to a minimum.

The previously reviewed SOI, Buried Layers, Ringed Transistors and Guard Ring techniques, which increase the energy deposition necessary to affect a transistor, also serve as hardening method for Single Event Transients by reducing the amplitude and duration of the transient.

Single Event Transients mainly affect analog and mixed-signal circuits, eventually causing Single Event Upsets in the digital parts of the latter. To reduce the amount and severity of SETs in the analog components of these systems, several approaches can be taken at different hierarchical levels of the circuit.

At design and layout levels, there are two main mechanisms to harden an analog

circuit against SETs, and they can be applied separately or combined to achieve a better performance. The first mechanism consists in increasing the amount of charge needed to create an SET in a node, the so-called critical charge. The second mechanism is based on the reduction of the charge collected by individual sensitive nodes.

Increasing the critical charge usually involves changing the design or layout of the circuit to increase the size of transistors and capacitors or increase the driving current and supply voltage. Among the strategies to increase the critical charge we can find the addition of low- or bandpass filters in critical nodes, which increases their capacitance and suppresses high-frequency SETs [86], increase the operating frequency (which involves an increase in the drive current) [86], or the insertion of decoupling resistors at certain points of the circuit to increase charging time constants, create low-pass filters in association with capacitors or absorb part of the voltage created by the SET. These concepts can be applied to a large variety of circuits, such as memory cells, digital latches or Voltage-Controlled Oscillators [87].

Decreasing the charge collected by the nodes can be achieved by layout design methods we have covered in previous sections. Introducing guard rings and taps in different regions of the design or using Silicon On Insulator techniques in the wafer may reduce the amount of charge collected by transistors and diodes, reducing their sensitivity to SETs. Other specific layout hardening techniques against SETs include Node separation and Interleaved layout. Node separation aims at reducing the chances of nodes other than the primarily struck node to collect part of the deposited charge. Although not very high, the accidentally collected charge can be significant and produce state changes. Fabrication technologies smaller than 100 nm are sensitive to this effect [88]. In Interleaved layout, sensitive nodes of the circuit are placed amidst pairs of less sensitive nodes to separate them and decrease the chances of charge collection whthout renouncing to high-density circuits [88].

Besides the two strategies involving the collected charge at circuit nodes, other common approaches can be additionally implemented in analog and mixed-signal circuits. These techniques usually involve changes in the architecture of the circuit to reduce the impact of transients or even mask their effects.

- **Analog redundancy**
  Reducing the impact of transient perturbations can be achieved through the addition of an arbitrary number of redundant instances of a circuit. The individual outputs are connected together through N parallel resistors at a common node, as in Fig. 2.19. As a result, a voltage transient in any of the redundant circuits is able to reach the common output, but its magnitude is divided by the number of redundant components added to the design.

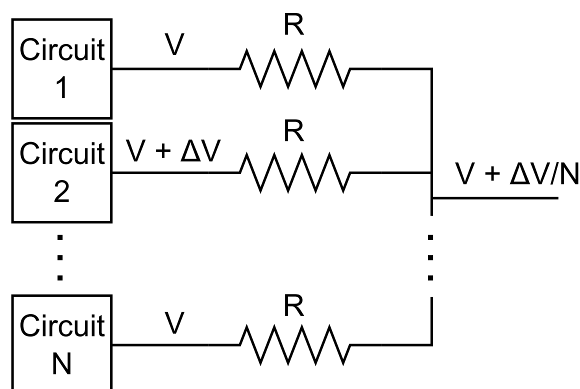


Figure 2.19: Analog redundancy architecture.

- **Window of vulnerability**
  In mixed-signal electronic circuits, Single Event Transients may be captured by memory elements provided they reach the input during the time frame in which the stored value is open for changes, usually during rising or falling edges of the clock signal. This time period is called the Window of Vulnerability. Reducing this time lapse contributes to reducing the amount of latched SETs (that are essentially transformed into Single Event Upsets). The Window of Vulnerability depends on the SET width and the technology of the latch, essentially of the setup and hold times, which constitute the sampling time. Thus, latches with shorter sampling times should be more robust to SETs, whereas hardening techniques able to reduce the length of the transient should also contribute towards reducing the window of vulnerability [89]. Additionally, lower clock frequencies should reduce the probabilities of a SET being latched into a memory element, since the fraction of time the latch is sensitive to SET with respect to the clock period is lower at lower frequencies.

- **Differential design**
  Differential design is a common-place technique used in high-performance analog electronic systems to mitigate noise. The output of the circuit is calculated as two separate results who are then differentiated. This way, noise affecting both branches in the same way is mitigated in the final output. This type of design cannot be used as it is in radiation environments, since a single particle hit would only cause one of the branches to fail and the error would not be mitigated after the differentiation. However, a modification in the layout of this kind of designs can be introduced to exploit charge-sharing and make the transient affect both branches of the differential pair, mitigating SETs. This modification is based on splitting every transistor in two devices and interleaving them with its counterpart from the other branch, placing the devices in the same well with the drains as close as possible to maximize the chances of sharing the charge deposited by an ion strike [90].

- **Pulse quenching**
  Charge sharing can also be used to decrease SET pulse-width in modern

technologies. The pulse quenching phenomenon has been reported as a naturally occurring mechanism in inverter chains used to study SETs, but can also affect other systems. This effect happens when a PMOS transistor in an inverter is affected by an ionizing particle, temporally changing the state at the output of the logic gate. The pulse reaches the next inverter, forcing a state change in its output. Before the state of the second gate is reverted to normal, charge deposited in the first PMOS diffuses to the second gate, forcing the state change to its normal operation in shorter time, leading to an SET pulse of shorter width than the original [91].

- **Reduction of high impedance nodes**
  Nodes with high impedance affected by a particle strike need longer times to recover the normal operation condition after the SET. Thus, reducing the impedance of sensitive nodes leads to shorter transients and the effects caused by SETs are less damaging in analog and mixed-signal circuits, where the transient has less probabilities of being stored in a memory element [92].

Specific SET mitigation techniques for digital circuits will be discussed in Section 2.6 as they are related to other digital error mitigation techniques.

2.5.E. SEU hardening

Hardening against Single Event Upsets is a central piece of this Thesis and we will cover it extensively in the next section.

2.6. Hardening against SEUs

Single Event Upsets affect all kinds of sequential digital circuits. Different hardening techniques devoted to the mitigation of SEUs have been proposed over the years, some of them can be generally applied, while others are specially devised for a purpose or circuit type. These hardening methods are usually applied at architectural level and most of them rely on some sort of redundancy to protect the stored data from corruption.

In this section we will cover the most common approaches towards Single Event Upset hardening in different systems that use digital electronics, from basic digital circuits used in larger blocks, to more complex components, such as memories, FPGAs or microprocessors. This section is the fundamental core of this Thesis; many of the hardening methods discussed here are the starting point of the research conducted in this Thesis.

2.6.A. Digital circuits

In order to harden digital basic components, several approaches can be taken. Applying a number of them concurrently provides a better error tolerance to the final design. Among

the general techniques we can distinguish between spatial and temporal redundancies. As for specific functions of the circuit, such as state machines or clock lines, other hardening methods have been proposed.

Spatial Redundancy

Spatial redundancy is based on the addition of hardware replicas of a sensitive module. The results calculated by these replicas can be compared to detect and even correct errors in the final result, by means of a comparison or a voting logic. The different spatial redundancy methods always propose a trade-off between the system's reliability and its area and power consumption.

- **Duplication With Comparison (DWC)**
  Duplication With Comparison, also called Dual Modular Redundancy (DMR), is the most basic hardening method of digital circuits. DWC is composed by the circuit that the designer wants to protect and a single copy of that same circuit. Their outputs are then compared and an error is raised if the two outputs differ [93]. Fig. 2.20 shows the configuration of a typical DWC. This technique only allows for error detection. Automatic correction cannot be performed, since a mismatch in the outputs does not give enough information to know which of the two circuits failed.

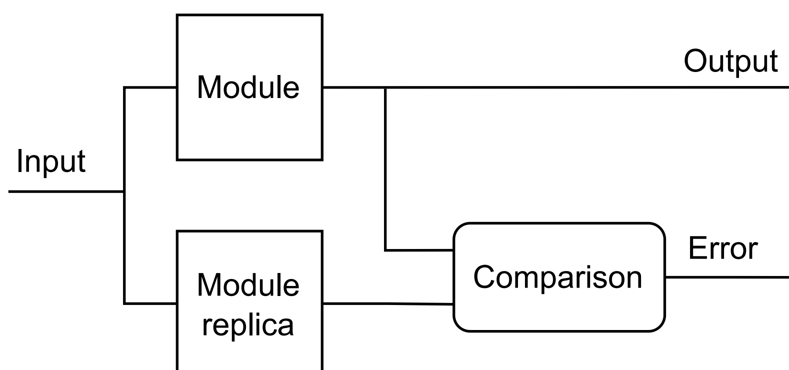


Figure 2.20: Basic structure of the Duplication With Comparison hardening method.

  Upon error, and depending on the application, two courses can be taken. If the application requires high throughput and does not care about a single calculation being lost, the faulty data can be discarded. On the contrary, the calculation may be retried if the data is important, but may be delayed.

- **N-Modular Redundancy (NMR).**
  To cope with the inability of DWC to correct errors, additional replicas of the module to protect can be added to the design. Instead of Dual Modular Redundancy, we call them N-Modular Redundancy. Among the many possibilities of NMR, the most popular one is Triple Modular Redundancy (TMR), which is the most efficient configuration

in terms of area and power consumption while offering high reliability and automatic error detection and correction.

The basic NMR schematic is composed by inserting two replicas of the design to protect and a voting logic. The voting mechanism of NMR is usually called "majority voter" because of how it works: between the results given by the modules, the voter chooses the most frequent one. In case a single module yields a faulty result, the rest will agree on the correct result and the error will be masked in the voting process. This approach has an obvious flaw: errors in multiple replicas will cause the voting to fail and an unexpected result will be chosen at the output. For the particular case of TMR, faults in more than one copy will trigger an erroneous correction. Fig. 2.21 shows an NMR design and the majority voting logic circuit.

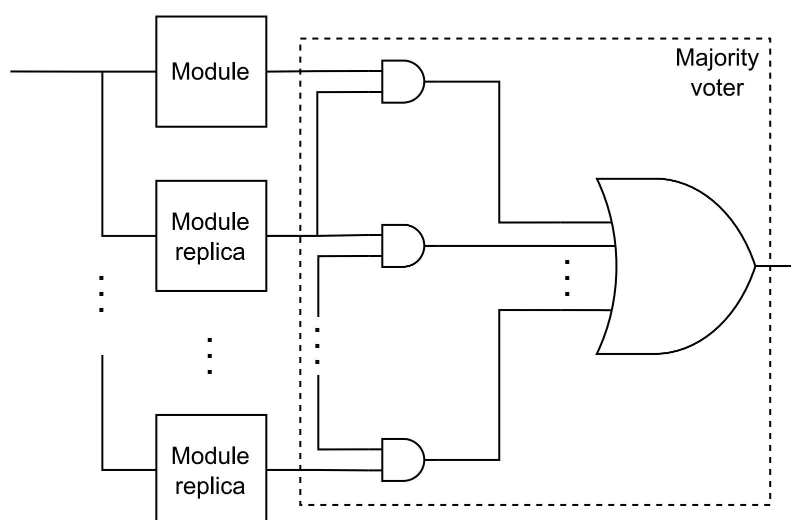


Figure 2.21: N-Modular Redundancy basic schematic.

N-Modular Redundancy can be applied at different granularity levels in the architecture of the circuit: from hardening individual memory elements, to whole blocks or components. For the sake of simplicity, and because it is the most common technique, we will just cover the Triple Modular Redundancy variations, but extrapolating the explanations to NMR should be straightforward [94].

– **Local Triple Modular Redundancy (LTMR)**
  Local Triple Modular Redundancy is implemented at register level. In this hardening technique, the combinational logic in the data-path is kept singular and the flip-flops are triplicated and voted with a single majority voter, as in Fig. 2.22.

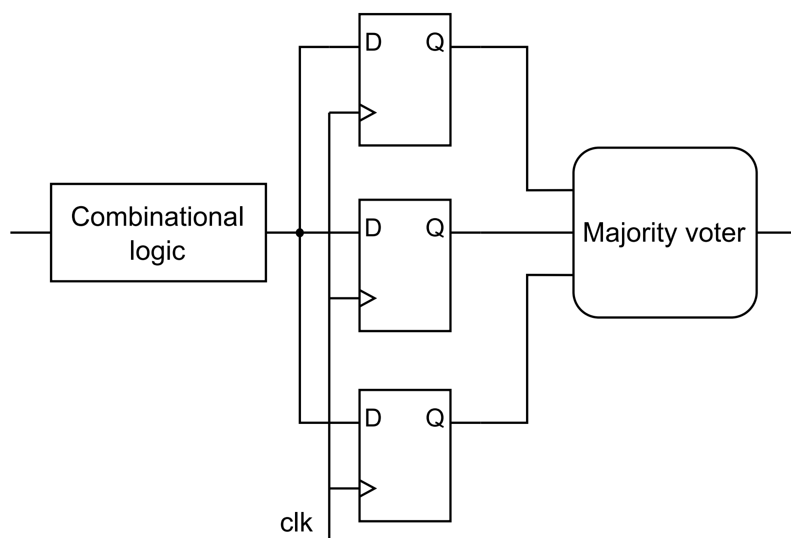


Figure 2.22: Block diagram of the Local Triple Modular Redundancy (LTMR) mitigation technique.

This hardening method is the most simple, but also the one that offers less protection. An error propagated in the combinational logic could reach the triplicated flip-flops and the error would not be corrected by the voter.

Radiation tolerant flash-based FPGAs such as Microsemi's RTG4 implement embedded LTMR in their flip-flops to decrease the sensitivity of the circuit to SEUs.

– **Distributed Triple Modular Redundancy (DTMR)**

  Distributed Triple Modular Redundancy is an improvement of the LTMR hardening technique, also implemented at register level. In this implementation, the combinational logic, flip-flops, feedback loops and voters are triplicated. This way, an error affecting one of the combinational logic data-paths would reach just one of the flip-flops and the error would be dismissed by the triplicated voting logic. Fig. 2.23 shows the implementation of the DTMR technique.

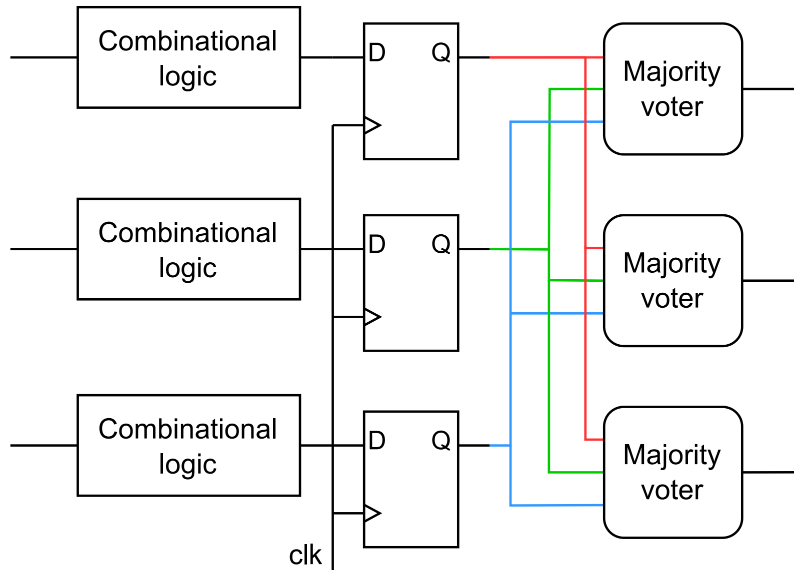


Figure 2.23: Block diagram of the Distributed Triple Modular Redundancy (DTMR) mitigation technique.

This hardening technique offers a significant improvement of the SEU hardness with respect to the LTMR technique. However, this mitigation technique does not completely immunize the circuit against SEUs: errors in clock and reset lines may cause the system to fail. Additionally, this technique greatly increases the resources used by the design, which also increases the chances of having Common-Mode Failures and other uncorrectable errors.

– **Global Triple Modular Redundancy (GTMR)**
  Global Triple Modular Redundancy is an improvement of the DTMR hardening technique in which the clock and reset lines are also triplicated. This way, the hardening system is able to mask errors affecting the inputs that were common to the triplicated flip-flops in the DTMR and were able to bypass the hardening. This schematic is shown in 2.24.

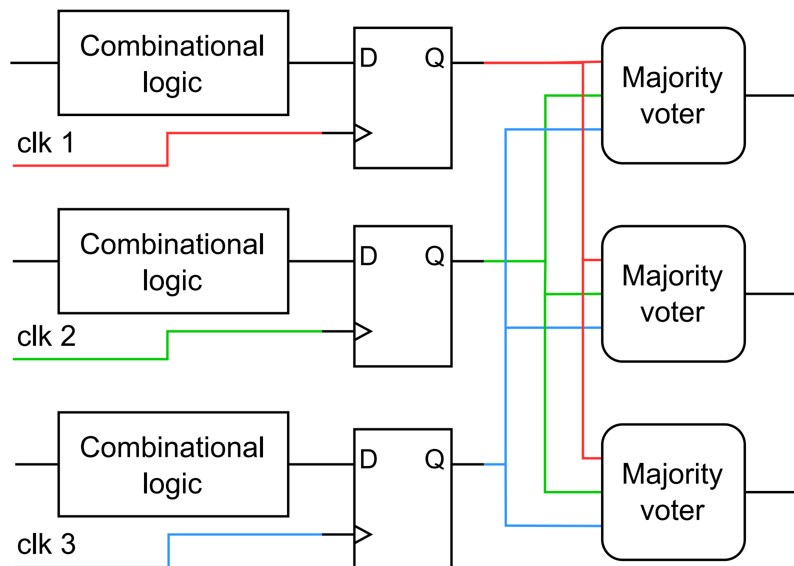


Figure 2.24: Block diagram of the Global Triple Modular Redundancy (GTMR) mitigation technique.

This technique offers a strong hardening against SEUs, but also features some implementation difficulties. Besides the obvious increase in the resource and power consumption, ensuring that the three clock lines are perfectly synchronised is difficult. Clock skew between the three clock domains must be kept below the shortest routing delay between adjacent flip-flops to be sure the same data is latched by the triplicated flip-flops at any given moment.

- **Block Triple Modular Redundancy (BTMR)**

  Block Triple Modular Redundancy is a TMR technique applied at a higher hierarchical level than the previous techniques. BTMR is specially devised to harden IP blocks or modules over which little control can be exerted, such as microprocessors implemented in hardware. In BTMR, whole blocks are triplicated and their results can be voted using a majority voter to improve their robustness. Fig. 2.25 shows a basic BTMR implementation.

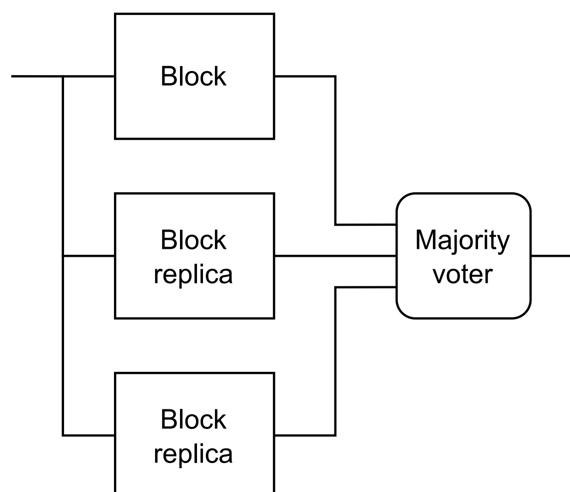


Figure 2.25: Block diagram of the Block Triple Modular Redundancy (BTMR) mitigation technique.

  In practice, BTMR consumes near as many resources and power as DTMR, but is much easier to implement. However, designs hardened using this technique must be reset or power-cycled frequently to avoid error accumulation in the logic inside the replicas. The probabilities of having Common-Mode Failures increase with error accumulation, decreasing the robustness of the design.

  Implementing BTMR at even higher hierarchical levels is a common practice in aerospace applications, where whole systems can be replicated to guarantee the availability of critical systems.

- **Diverse Triple Modular Redundancy**

  Diverse TMR is a variation of TMR in which the redundant replicas of the circuit to protect compute the same algorithm (with the same inputs and outputs), but implemented in a different manner. This way, the three circuits have diverse responses to similar faults, and may even need less clock cycles to finish the computation or need less resources. This technique has shown promising results when compared to BTMR designs implemented in FPGA [95]

The previously reviewed techniques, specially DTMR and BTMR, are considered reliable standard techniques for space applications and have been included in critical systems since the discovery of radiation effects on electronics. Traditional space trends favor mitigation techniques that assume less risks, even at higher manufacturing or design costs. Nevertheless, perfect error correction is not always needed, because of the lesser criticality of a process or because exchanging robustness for less consumption is preferred.

For those cases in which errors may be tolerated under certain circumstances, less consuming alternatives to NMR have been explored. Approximate computing is a very interesting field for this purpose, since approximate calculations can usually be translated in the hardware domain into less consumption of power and resources.

- **Approximate Triple Modular Redundancy (ATMR)**
  The Approximate TMR technique consists in hardening a given digital circuit by adding two copies of the design that are not exactly equal to the original. The Approximate copies yield the same results as the original circuit just for a fraction of the inputs in the input space. The approximations can either expand the number of cases in which the circuit yields a logic 1 (over-approximation) or a logic 0 (under-approximation). The usual ATMR implementation uses an over-approximation and an under-approximation circuits alongside the target circuit to establish a comparison and detect and correct errors in the circuit. The main advantage of this approach is that the approximate circuits need not have the whole domain of the original circuit and may even be implemented with less inputs, thus, with less logic. This technique, however, only allows mitigation at bit-level, so it is a good method for control logic or state machines, but it is less effective and more difficult to implement in complex data handling algorithms [96].

- **Reduced Precision Redundancy (RPR)**
  In the Reduced Precision Redundancy technique, the circuit to protect is hardened using two identical circuits that compute the same function, but with less precise data. A smart comparison voter based on an acceptable error threshold allows for errors in the Full Precision circuit to be approximately corrected using Reduced Precision results. The amount of bits discarded in the calculations of the Reduced Precision modules leads to a trade-off between the resources needed by the mitigation circuits and the quality of the corrected results. This technique achieves significant reduction of resources and it is easy to implement in Digital Signal Processing (DSP) algorithms [97] [98]. Several implementations have been proposed over the years to harden different DSP algorithms using RPR, such as basic arithmetic operations [99], [100], [101], Infinite and Finite Impulse Response Filters [102], [103] for different space-related applications, or Fast Fourier Transform architectures [104]. Other architectures increasing [105] or decreasing [106] the amount of replicas used in the RPR have been proposed to increase reliability or to reduce resources. The simulations

and fault injection experiments carried out in these studies confirmed that the RPR technique is able to reduce the power and resource consumption of the hardened design compared to other hardening techniques while maintaining a high error detection and correction rate. In this Thesis we will discuss extensively about Reduced Precision Redundancy, the influence on its correction capabilities of certain parameters such as the architecture employed or the bit-width chosen and the sensitivity of RPR-hardened designs under the effects of radiation, which has not been examined in previous studies.

Temporal Redundancy

An alternative approach to the basic TMR techniques is the concept of temporal redundancy. Temporal redundancy is based on parallel computations performed with a certain delay between them, either to reduce the resource overhead or to improve robustness:

- **Offset Dual Modular Redundancy**
  The Offset DMR technique exploits the temporal redundancy to improve the Duplication With Comparison concept and determine, under certain circumstances, which of the two circuits presents errors. This way, the DWC is in most cases able to select a correct output and not just signal the mismatch between the two results. The error detection mechanism shown in [102] benefits from how soft errors propagate through a pipelined Fast Fourier Transform to recognize mismatch patterns in the output samples and determine a correct result. This mitigation technique should be able to reduce the overhead caused by the mitigation to just two times the original size, instead of three, in the case of a TMR implementation. Nevertheless, it is based on certain mathematical properties of the FFT and may not be easy to port to other kinds of algorithms.

- **SET filtering TMR**
  Temporal redundancy can also be used to reduce the window of vulnerability of TMR-hardened flip-flops against Single Event Transients. By placing a small and different delay in the data or clock lines of two of the three flip-flops in the triplet, an incoming SET may be masked [41]. Fig. 2.26 exemplifies the two described SET filtering options.

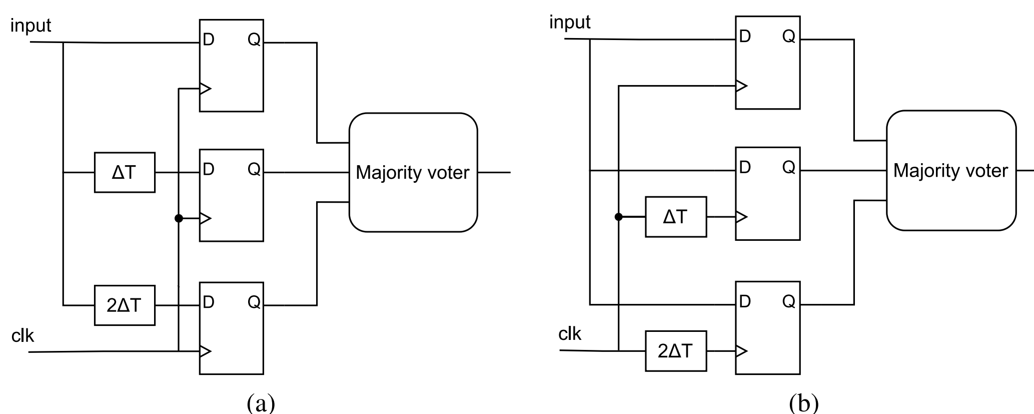


Figure 2.26: SET filtering TMR implemented in (a) data line and (b) clock line.

The delays are usually inserted as chains of inverter gates, which complicates manual insertion by the user. Implementation is usually performed by automatic tools through modifications in the synthesized netlist or embedded in the chip in the case of some radiation tolerant FPGAs with embedded TMR in their flip-flops [107].

Fail-safe FSMs

For the particular case of hardening Finite State Machines (FSM) there are two possible types of malfunctioning that have to be taken into account in the design. Specific measures to tackle them have been proposed:

- **Eliminate transitions to illegal states**
  The current state of an FSM is stored in a bit vector. An FSM with N different states would need a vector of 2^{N-1} bits; unless N is a power of 2, this means that there will be values in the vector that do not correspond to any valid state of the FSM. Because of this, it is important to make sure that an SEU affecting the current state register does not send the FSM into a state from which it cannot recover, a deadlock state. For this, the designer has to create transitions from illegal states to a known state and make sure these connections are not lost in the optimizing process done by the synthesizer. The "when others" and "default" clauses, in VHDL and Verilog respectively, help in creating fail-safe transitions, whereas specific vendor attributes can be added to the signal to prevent them from being optimized away.

- **Prevent transitions to incorrect legal states**
  A bit-flip in the state register can also cause that the FSM is forced into a valid state, but an unexpected one. Since FSMs are key parts for the correct functioning of the system and do not usually need many resources, hardening the state register using TMR is a simple and reliable way to prevent wrong behaviours of the FSM. Other options, such as encoding the state register with an Error Detection And Correction (EDAC) [108][109] code or using rad-hard components to store this data are viable options. Regarding

codifications for the state register, it is worth mentioning the one-hot codification, in which every valid state is represented by a bit in the register and so, only a bit is at high logical level ("hot") at a time. Single bit flips in the register would cause it to represent an illegal state, that can be corrected using fail-safe transition techniques.

Other hardware resources

Other specific hardware resources have to be considered when implementing mitigation techniques for digital circuits. In particular, the clock an reset lines, which are fundamental in any sequential circuit.

As a general rule, using special radhard cells provided by vendors is a good measure. Cells with high drive or high fan-out strength to distribute asynchronous signals such as the reset or clock lines help reducing the chances of an SET affecting the system. These cells are made with transistors of larger sizes, which increases the critical charge needed to create a perturbation in the line.

Asynchronous lines suffer degradation due to radiation in two ways: increased jitter and transients that can be mistaken for rising or falling edges. These two effects are specially dangerous in clocks and set/reset signals of digital circuits. To mitigate these errors, besides the radhard cells, triplicating the lines, although expensive, may be of help. Increasing the critical charge in the clock distribution network, particularly in the first stages, helps in reducing the chances of SET propagation along the line [110].

2.6.B. Memory elements

Memory elements are a very important part in digital circuits and they are particularly sensitive to Single Event Upsets. NMR techniques can be applied to a small number of memory elements, such as registers, but they are not so efficient to harden large amounts of arrays of memory cells, because of the associated area and power overheads. Techniques other than spatial redundancy can be applied at different levels of the design, even simultaneously, to increase reliability. Among the most notable mitigation techniques for memories, we find radiation hardened cell designs, the addition of error codes, spatial redistribution of individual bits to avoid MBUs, or the employment of external supervisors, which we describe in the following sections.

- **Hardening memory cells**
  The basic Static RAM cell is implemented using 6 transistors: four of them form a cross-coupled pair of inverter gates that stores the input value, and the two remaining transistors are used to enable the read and write operation. Fig. 2.27 shows the configuration of a 6-transistor (6T) SRAM cell, capable of storing a single bit of information.

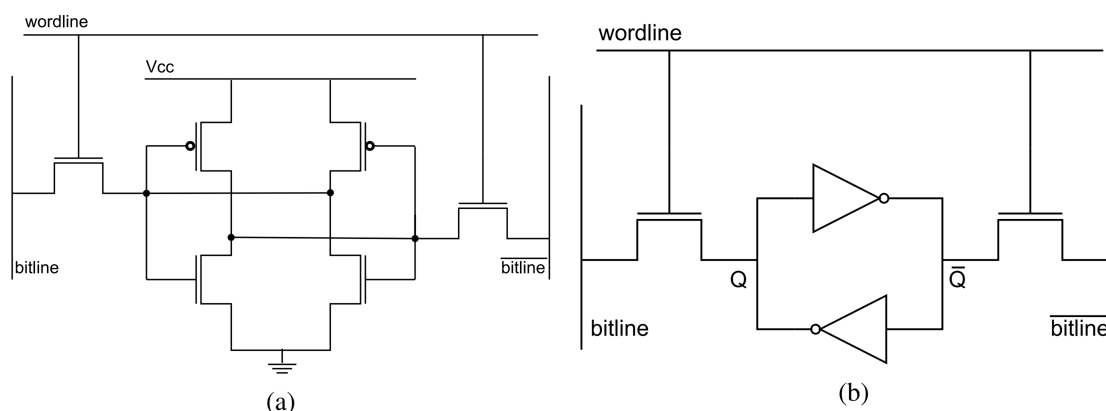


Figure 2.27: 6T SRAM cell: (a) transistor architecture, (b) simplified view of the same architecture.

Based on this architecture, some modifications can be applied to the cell to improve its radiation performance.

As seen in Fig. 2.28a, resistors can be placed in the cross-coupling paths to attenuate the impact of transients and avoid storing short changes in the logical state of the signal [111]. By placing a capacitor between the cross-coupling lines, as in Fig. 2.28b, the critical charge of the cell can be increased to improve its hardness. Although these techniques increase the immunity to SEUs of memory cells, they also impose a high speed penalty and area overhead [112].

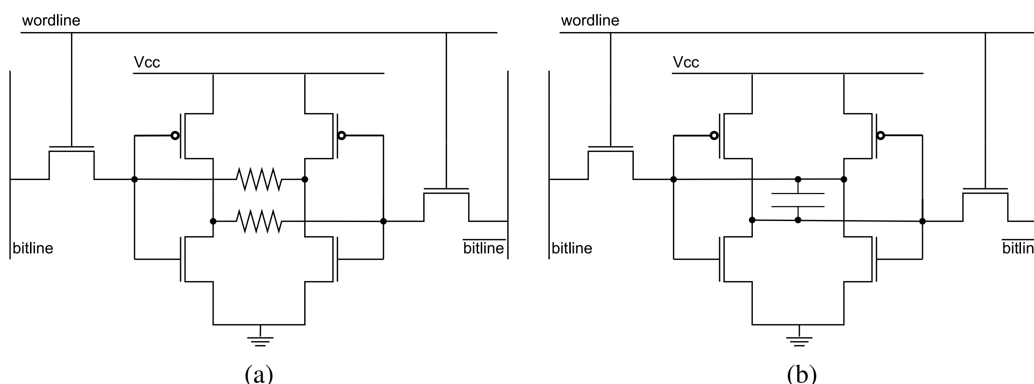


Figure 2.28: (a) Resistive and (b) capacitive mitigation techniques.

Alternative approaches based on the addition of redundant transistors, have been proposed and successfully implemented in a large variety of space components and missions and are part of several radhard libraries. These approaches usually come with an at least 100% overhead, but little to none speed penalty and only a small power consumption overhead. Besides, because transistors are the only redundant components added, the technique can be easily scaled to smaller sizes. The redundant transistors in these techniques are placed in feedback paths to be used to restore wrong values caused by SEUs in individual transistors. How the transistors are interconnected is the main difference between these designs.

Some examples of this kind of techniques include the Heavy-Ion Tolerant (HIT) memory cell [113] or the Dual Interlocked CEll (DICE) [114], shown in Fig. 2.29.

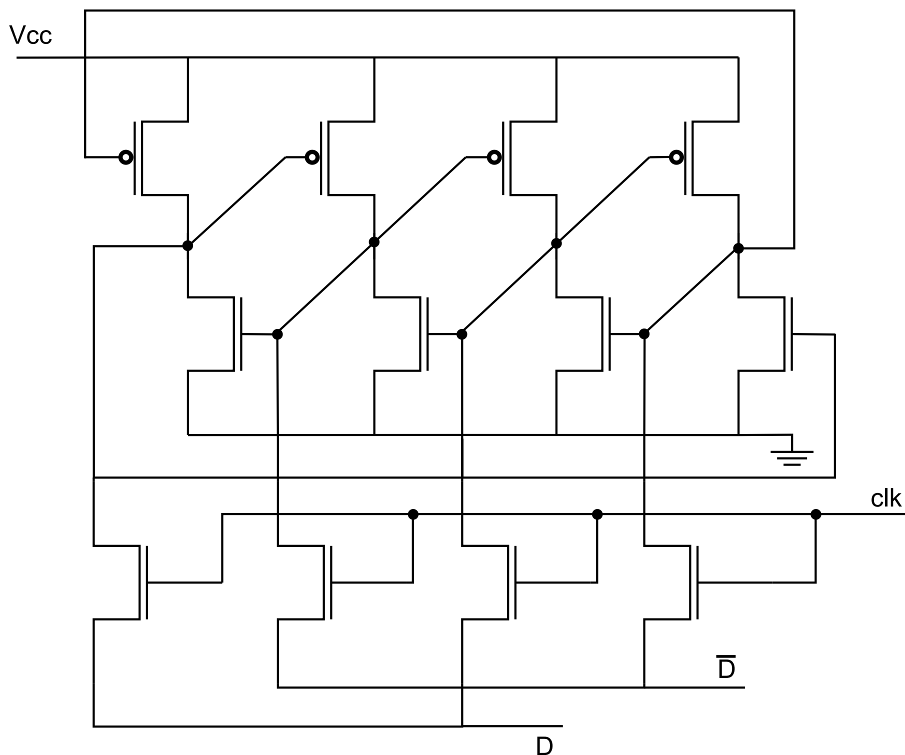


Figure 2.29: Dual Interlocked Cell (DICE) architecture.

- **Error Detection And Correction codes**

  Error Detection And Correction (EDAC) codes are error mitigation techniques capable of detecting and correcting errors in transmitted or stored data through the addition of redundant bits to the data [115].

  All these codifications work in a similar way. First, the data to be protected is processed to generate a unique code associated to the information. The data and the code are then stored in a memory or sent to other component through some media. When the data and the code are retrieved, the decoding algorithm is able to detect the presence of errors thanks to the redundant bits in the code and, in some cases, is capable of locating and correcting a certain amount of them.

  EDAC codes can be roughly divided in two categories: those based on performing mathematical operations on the data and those based on parity. The encoding algorithms based on operations, such as the Solomon-Reed code or Cyclic Redundancy Codes, are normally just used in communication processes in space applications because the encoding and decoding operations are complex and introduce high time overheads, which is not desirable when accessing a memory.

  Parity-based codifications count the number of bits in the logical state '1' (or '0') in some parts of the data to protect, and they add parity bits whose value depends on the

parity of the number of '1's counted. The most simple parity code adds a single parity bit; this allows for single-bit error detection, but not correction.

Hamming codes represent an improvement from the basic parity algorithms, as they are able to detect up to two erroneous data bits and correct single bit errors. The general idea behind Hamming codes is the addition of parity bits that encode the parity of certain positions in the data word to protect. Depending on the length of the data word, a different amount of parity bits is added. Hamming codes become more efficient in terms of overhead with larger words. For instance, a Hamming (7,4) code uses 3 parity bits to protect a 4-bit word (using a total of 7 bits), while a Hamming (255, 247) needs 8 parity bits to protect a 247-bit data word (a total of 255 bits).

To understand the encoding process carried out in Hamming codes, we will use the basic Hamming (7,4) [116] as an example, since other Hamming configurations work in a very similar manner. This codification adds 3 parity bits to protect 4 bits of data. The parity bits are placed in the first, second and fourth positions of the word (the positions that are a power of two) and the data bits are in the third, fifth, sixth and seventh positions (the remaining positions). Each parity bit protects (counts the number of '1's) the bits whose position ANDed with the position of the parity bit gives a non-zero result. This method can be expanded to protect larger data words by adding extra parity bits and becomes more efficient at larger sizes in terms of overhead needed. Table 2.2 helps understanding the encoding process. The X symbols mark the positions checked by each parity bit.

Table 2.2: HAMMING (7,4) PARITY TABLE.

Bit position		001	010	011	100	101	110	111
Data bits		p1	p2	d1	p4	d2	d3	d4
	p1	X		X		X		X
Parity bit coverage	p2		X	X			X	X
	p4				X	X	X	X

To check for errors, the parity bits can be recalculated upon retrieval of the data and compared with the received ones. By checking which parity bits differ from the theoretical parity bits the position of faults can be determined. The encoding and decoding process is easily performed as a series of matrix multiplications. The Hamming (7,4) algorithm is able to detect up to 2 errors in the 7-bit data word or correct one bit errors. By adding a higher proportion of parity bits, a higher amount of errors can be detected and corrected. The Single Error Correction - Double Error Detection (SECDEC) algorithm adds an extra parity bit to be able to correct a single error or detect 2 errors in the word.

- **Scrubbing**
  Scrubbing is a refreshment operation performed in embedded memories whose contents

are not overwritten often enough, to prevent error accumulation. In this operation, the contents of the memory are regularly read, checked for errors and corrected using EDAC approaches when an error is detected. A special application case is the configuration memory in FPGAs. As the FPGA configuration is static, it is possible to refresh the configuration periodically from a correct source without the need of checking it.

There are two scrubbing approaches that can be used: the deterministic approach, that scans the whole memory word by word, and the probabilistic approach, in which data words are checked only when they are accessed (read or written).

In deterministic scrubbing, operations should be performed at a sufficiently high rate, around ten times faster than the expected SEU rate of the memory, to avoid error accumulation, since a high number of faulty bits in a data word may corrupt the data past the hardening limits of the EDAC code implemented.

Probabilistic scrubbing assumes that every word in the memory is accessed uniformly, which is not often the case for real applications. Hybrid deterministic-probabilistic or hierarchy-based partial scrubbing approaches should be considered when a compromise between reliability and speed is desired [117] [118].

- **Bit interleaving**

  The bit-interleaving method, also called bit-scrambling, is a mitigation technique dedicated to reduce the sensitivity of memories to Multiple Bit Upsets. As we saw in Section 2.2.C, MBUs are a special kind of errors that appear in memories when a single particle hit upsets more than one bit.

  A simple yet effective method to reduce the sensitivity of a memory to MBUs is to rearrange the words and bits in the memory so that the logical structure of the memory is different from the physical structure, avoiding logically-adjacent bits being also physically adjacent. Thanks to these changes, particle strikes affecting adjacent memory cells should only create single-bit errors in different words spread across the memory, which could be corrected by the previously described Error Detection and Correction Codes and the Scrubbing techniques [119]. Fig. 2.30 illustrates an example of the changes needed to apply the bit interleaving technique.

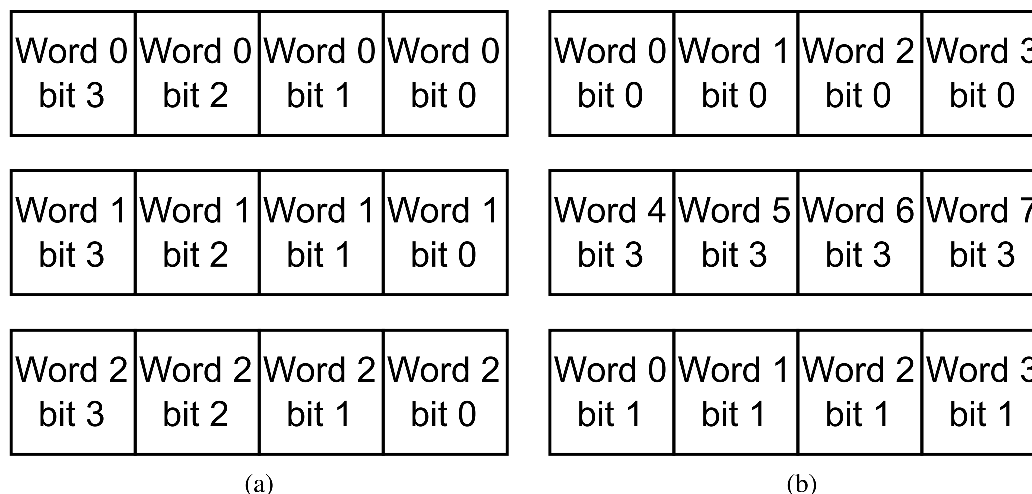


Figure 2.30: SRAM memories in which (a) the logical structure is the same as the physical structure and (b) the logical structure has been interleaved to mitigate MBUs.

2.6.C. Hardening complex devices

Complex devices are commonplace in space applications for their good performance and processing capabilities, specially in modern times, since the adoption of New Space methodologies in which the usage of COTS is increasing.

Complex devices are characterized by encompassing multiple functionalities and systems in the same chip. This heterogeneity presents challenges not only to test the systems, as we saw in Section 2.4.B, but also to harden them. The hardening approach for these systems must start by a careful analysis of the device and then apply individual hardening techniques to every subcomponent, paying special attention to the most critical functions. Nevertheless, this strategy is usually difficult, if not impossible because of vendor restrictions in COTS components: the characteristics of some parts of the system may not be publicly disclosed to the user and other parts may not be modified, which complicates finding workarounds to harden them. That is why implementing global supervisors able to reset the system in case of an event causing unrecoverable malfunctioning of the device is key for complex devices.

In this section we will cover specific techniques to harden FPGAs and microprocessors and we will briefly discuss how to implement mitigation techniques in MultiProcessor System-On-Chip (MPSoCs), powerful devices including both an FPGA and a multicore processor in the same chip.

Hardening FPGAs

FPGAs are configurable digital devices with high versatility, which makes them very useful for space applications. The implementation of circuits in the programmable logic of the FPGA is done through the definition of interconnections between the preexisting

hardware resources of the FPGA. The circuit to be implemented is usually described using some sort of Hardware Description Language (HDL), such as VHDL or Verilog, and then a synthesis software transforms this design into a configuration bitstream, that defines the connections to reproduce the behaviour of the circuit in the logical resources of the FPGA.

Therefore, FPGAs have two layers of resources: the configuration memory, in which the configuration bitstream is stored, and the application logic, which includes Look-Up Tables, Flip-Flops and memory blocks. Both the configuration memory and the hardware resources of the FPGA might be subject to radiation. Depending on the technology used to implement programmable elements, there are different types of FPGA, each of them exhibiting a different behavior under radiation. These three types of FPGA are detailed below.

- **Antifuse FPGA**
  The interconnections between the resources are permanent; thus, reconfiguration is not possible. This type of FPGA has been traditionally used in space applications because of its inherent radiation hardness: having a fixed configuration, SEUs cannot affect the configuration resources and only logical resources are sensitive to radiation.

- **Static RAM FPGA**
  SRAM-based FPGAs store the configuration bitstream in an SRAM memory. This makes the FPGA reconfigurable at any time, but also quite sensitive to SEUs in the configuration layer, which can affect the behaviour of the system. Faults in the configuration memory cannot be corrected by a simple reset, so other countermeasures must be taken. SRAM memories are volatile, so the configuration is lost if the power is cut from the device.

- **Flash FPGA**
  The configuration bitstream is stored in flash memory in this kind of FPGAs. As was the case of SRAM-based FPGAs, the configuration layer is also sensitive to SEUs, but flash memory cells are far less sensitive to SEUs than SRAM cells. On the other hand, TID particularly affects Flash transistors [120]. Faults in the user logic are the main source of faults in this kind of devices. [121].

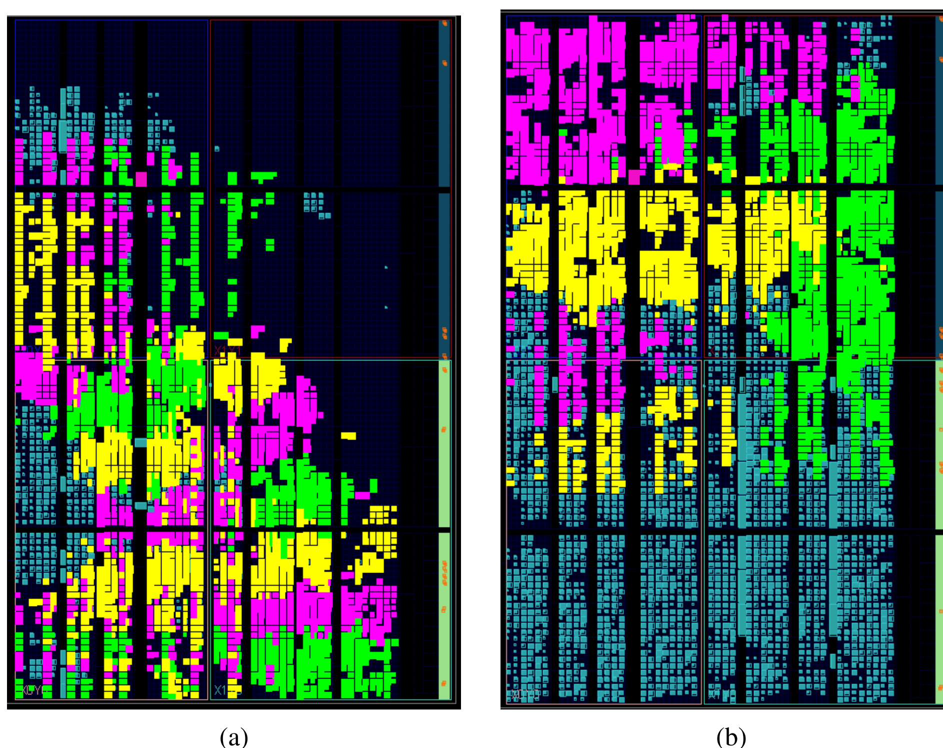
The FPGA technology determines which hardening strategies must be followed in order to reduce the sensitivity of the design against SEEs. These are some of the mitigation techniques that must be considered when designing circuits protected against radiation:

- **Digital/mixed electronic hardening**
  The techniques described in Section 2.6.A, that are specially conceived for digital circuits, are good hardening methods for designs implemented in FPGA. However, there are some aspects that must be considered when implementing some of these techniques in FPGAs, specially for SRAM-based devices.

As previously explained, the DTMR technique is quite effective against SEUs, being able to correct most single errors and prevent error propagation. Although very effective in error masking, this technique still has one flaw: errors affecting more than one of the replicated data paths may break the correct functioning of the TMR voting strategy. While this might not be so probable in ASIC or antifuse implementations (it would require upsetting two adjacent transistors belonging to different data paths at the same time), it is much more frequent in SRAM FPGAs. Upsets in the configuration memory distort the behaviour of the circuit and this may affect two redundant data paths and the fault may scape the mitigation.

The reason behind this is that due to the frequent interconnections between the three data paths in DTMR implementations, the redundant copies cannot be physically separated when placed in the FPGA fabric.

For SRAM-based circuit implementations, the Block TMR technique may reduce the occurrence of these Common-Mode Failures. A careful placing strategy might spread the redundant modules across the FPGA fabric, reducing the interconnection points and, thus, the probability of uncorrectable errors in the final output of the circuit [41]. Fig. 2.31 shows a comparative of a design implemented in FPGA using the DTMR and the BTMR hardening techniques. It is apparent from the figure that the three domains in the BTMR implementation are more clearly defined and could be eventually separated in the FPGA.



(a) (b)

Figure 2.31: Resource placement for the same design hardened using (a) DTMR and (b) BTMR. In yellow, pink and green, resources allocated for each TMR domain.

The sensitivity of DTMR designs against CMFs can be lowered by splitting the combinational logic blocks in various parts and inserting additional voting steps in between. While this can help correcting cross-domain faults, increasing the amount of logic may also increase the overall sensitivity of the circuit.

- **Place and Route techniques**

  Among the configuration bits used by a design implemented in an FPGA, the vast majority is devoted to route the resources, establishing the necessary connections between them to carry out the desired operation. The rest of them configures the Combinational Logic Blocks or the Look-Up Tables that define the results of combinational operations, the input signals of Flip-Flops, the internal routing of multiplexors and the configuration of other specialized blocks, such as multipliers, when available. For this reason, taking care of the correct placement of the replicas in a TMR design is important to avoid sharing resources between the three domains of the TMR as much as possible.

  To this end, some Placement and Route tools and techniques have been developed with successful results.

  The Reliability-oriented place and Route Algorithm (RoRa) [122] is a tool to perform automatic triplication, placement, and route of an unhardened design. The algorithm works in such a way that prevents routing data lines belonging to different TMR domains from being routed in the same interconnection nodes. This way, an error affecting the configuration bit of an interconnection node would not cause faults in multiple TMR domains, just in one of them, which should be corrected by the TMR hardening.

  Other techniques focus on the resource placement to reduce the amount of resources. The FPGA fabric is usually divided in square blocks called slices, containing a certain amount of diverse resources, with data paths separating them from each other in horizontal and vertical grids. The Incremental Placement technique modifies the placement of said slices to separate resources from different TMR domains in different slices. The Striping technique works in a similar way, but restricting the three TMR domains to different and separated columns of slices, creating a "striped" pattern in the placed resources [123].

  All of these techniques require a deep knowledge of the internal architecture of the FPGA, which is not usually provided by the vendor. The automatic tools that implement these Place and Route cannot be as efficient as the one developed by the vendor, and, as a result, the hardened designs may not be able to fulfill the timing requirements.

- **Scrubbing and reconfiguration**

  The previously described techniques work towards hardening the user logic (the designed circuit implemented in the FPGA), and reducing the sensitivity of the configuration memory to CMFs. However, SEUs in the configuration memory are

considered semi-permanent, since the wrong value will remain until the content of the memory is overwritten.

Periodic scrubbing of the configuration memory in FPGAs is mandatory to avoid permanent errors in the user logic. Scrubbing can be performed either on the whole memory or, if the scrubber allows it, on part of it. The advantage of partial scrubbing is that a detected error can be corrected faster and the frequency of the scrubbing is higher. Partial scrubbing requires that the design is hardened using Block TMR and that the voting logic locates the fault in one of the blocks. Then, the scrubber can read and check just the portion of memory corresponding to the affected redundant block, not the whole design [118] [124].

We define reconfiguration as overwriting the whole configuration memory of the FPGA using a safely stored bitstream. In SRAM-based FPGAs, whose configuration memory is volatile, a reconfiguration can be easily carried out by a simple power-cycle. Partial reconfiguration can be used to overwrite part of the contents of the configuration memory or to dynamically modify the placement of resources to unoccupied areas of the FPGA in case of a permanent error in a hardware resource [125].

Hardening microprocessors

Microprocessors, being complex devices, can suffer the effects of radiation in different ways. For this reason, numerous hardening techniques have been proposed and applied to harden them. Again, RHBP techniques apply in most cases to harden the low-level hardware that constitute the microprocessor, but for this work, we will cover RHBD approaches, which can be applied to COTS devices.

Radiation Hardening By Design of microprocessors can be attempted in two complementary ways: spatial redundancy, in which the hardware of the microprocessor is modified to improve its hardness, and temporal redundancy, which involves modification in the application running in the microprocessor to correct errors. For spatial redundancy implementations, the whole microprocessor, or parts of it, are replicated a number of times to allow error detection and correction. Other spatial redundancy techniques such as EDAC codes or error scrubbing can also be applied to the various memories and registers of the microprocessor. In temporal redundancy techniques, parts of the program running on the microprocessor are replicated and their results are compared to detect soft errors.

Additionally, the utilization of external supervisors that monitor certain parameters of the program and the microprocessor can be of help to detect and correct errors.

Replicating the entire microprocessor or parts of it and comparing the results of the redundant copies is the base of the spatial redundancy mitigation strategies for microprocessors.

- **N-Modular Redundancy**
  As for any other digital device, N Modular Redundancy can be implemented at different levels of the architecture of the microprocessor to mitigate errors.

  The basic Duplication With Comparison, usually called Duplex in this context, and the Triple Modular Redundancy, usually called Triplex, are popular implementations despite the lack of correction capabilities of the first and the large overhead introduced by the second alternative.

  Double Duplex is a variant of the Duplex architecture in which two Duplex modules are implemented, one master and one slave. Whenever a fault causes the microprocessors in the master module to disagree, the slave module is used to calculate a new result while the master module is reset [126]. Slave modules can be kept running at the same time as the master slave (hot redundancy) or shut down until a fault appears (cold redundancy). Hot spares provide faster responses upon error, but are a high power overhead for the system.

  Several variation of NMR systems can be implemented by adding cold redundant subsystems. Examples of this could be the Double Triplex, a TMR system and a spare TMR switched-off, or the Quadruplex, a TMR system with a spare microprocessor on cold redundancy [127].

  For all of these N-Modular Redundant systems, the voting step is performed only at certain points of the application running on the processors, usually just comparing the final outputs of the algorithm. This is called macro-synchronization. The individual components of a macro-synchronized system may take a slightly different time to complete the task due to differences in the fabrication process or to the ageing of the components subject to radiation, but this is tolerated by the voting process.

- **Lockstep**
  Lockstep is a technique based on micro-synchronization of two identical microprocessors running the same software at the same time. The execution of the program is synchronized at instruction level by an external hardware checker that compares the state of the two processors every clock cycle and raises an error if a mismatch occurs [128] [129].

  In lockstep, both microprocessors have access to read the shared memory, but only the master processor can perform write operations.

  Due to technology scaling, micro-synchronization is becoming less feasible: exact synchronization of the clock signals for both processors is difficult and asymmetric TID effects on the components of the master and slave processors also produces timing mismatches [127].

  Several lockstep solutions based on softcore processors (microprocessors implemented in FPGAs) have been proposed over the years because of the higher flexibility they provide, compared to the commercial hardcores [130] [131].

To compensate for the large area overhead caused by spatial redundancy, temporal redundancy can be seen as an alternative. The idea behind temporal redundancy is to execute the same operation a number N of times in a single piece of hardware and then compare the results to discard errors. For N equal or higher than two, N-Modular Redundancy allows error detection and correction [132]. Temporal redundancy in microprocessors can be applied at different hierarchy levels:

- **Redundancy at instruction level**

  To implement temporal redundancy at instruction level, the data structure of the original program has to be replicated N times. Then, every memory read, memory write and arithmetic operations must be replicated, each of them affecting its corresponding data structure. After each instruction has been executed N times, the data structures must be compared to check for consistency. For $N > 2$, majority voting can correct any inconsistencies. For $N = 2$ the checker routine forces a re-execution of the instruction to determine the correct result.

  Optimizations performed by compilers may inadvertently remove the redundant computations, so the designer is advised to disable optimizations when hardening high-level code. In case high performance is needed, it is recommended to introduce the replicas in the intermediate code generated after the optimizations.

- **Redundancy at task level**

  Temporal redundancy at task level is similar to redundancy at instruction level, but applied at a higher hierarchical level. Tasks in an application can be divided in three types: data acquisition, data processing and data presentation. As in instruction level redundancy, data structures and tasks are replicated N times and a consistency check is performed after each replicated task execution.

  In this case, a fault in a task execution takes longer to be detected, which can cause the fault to propagate further and induce errors in more than one execution. To avoid this effect, special care must be taken so that each data structure replica is confined to the same area of the memory, avoiding overlaps between replicas. Task executions must also be prevented from accessing data structures other than their own.

- **Redundancy at application level**

  At the highest granularity, we find redundancy at application level. A basic version of this redundancy can be implemented by just repeating the execution of the whole application a number of times and comparing the individual results of the each execution to check for errors.

  A different way to implement this hardening method has been proposed in [133]. For this, an operating system is run on a single microprocessor and two or more virtual machines are created to execute whole applications. A hypervisor manages the memory and resources of the virtual machines so that there are no interferences between them and performs the comparison of the outputs of each redundant application.

It is worth noting that these temporal redundancy techniques are good for correction of transient errors, but semipermanent errors in hardware components may surpass their mitigation capabilities: an error in the hardware may cause all the calculations to be faulty in the same point, rendering the majority voting useless. This is the reason why temporal redundancy is not recommended for softcore microprocessors implemented in SRAM-based FPGAs, or any other circuit implemented in SRAM-based FPGAs.

Temporal redundancy methods introduce a high execution time overhead in the hardened system with the idea of reducing the hardware overhead at a minimum. While this might be beneficial for the cost of the system, temporal redundancy might not fulfill the requirements of the mission for time-critical tasks. Approximate computing techniques can be exploited to reduce the execution time and power needed by the redundant calculations performed in systems hardened using temporal redundancy [134]. Among the most popular approximate computing techniques, we find bit-width reduction (which works in the same manner as Reduced Precision Redundancy), floating point to fixed point conversion in the redundant mathematical operations, or code perforation, in which unimportant parts of the code are identified and removed. Loop perforation is a good example of a code perforation technique. In loop perforation, the number of iterations in a loop is reduced to shorten the execution time. As was the case of approximate hardening techniques for digital circuits, approximate computing proposes a trade-off between the accuracy of the result and the execution time needed to calculate it. By applying approximate computing techniques to the replicated calculations of a temporal redundant system and adding a voting logic that tolerates some uncertainty, good correction results can be achieved at a lower computational cost [135], [136] [106].

Besides the error mitigation techniques to correct SEUs, other subsystems should be added to mitigate faults in the microprocessor that may disturb its normal operation, such as SEFIs or SEL.

- **Watchdog**
  Watchdogs are timers that perform a certain action if their maximum count value is reached. To avoid triggering the action, the system has to reset the timer periodically before it reaches the end of count. Watchdogs can be implemented internally in the microprocessor to monitor the correct functioning of a task, or externally to monitor the responsiveness of the whole microprocessor [137].

  Using watchdogs, tasks can be retried or the whole system can be restarted if the functioning of the application running on the microprocessor is slowed down or blocked by a Single Event Functional Interrupt.

  Soft resets, restarting the program running on the microprocessor, and hard resets, also called power-cycles, because they consist in turning-off the power of the chip, are in some cases the only way to recover from the effects of SEFIs.

- **Latching Current Limiters**
  External Latching Current Limiters should be added to the power line of a

microprocessor under the effects of radiation for two reasons. First, to protect the device against Latch-up, as we saw in Section 2.5.C. And second, because the basic LCL circuit can be modified to cut off power from the microprocessor on the command of an external supervisor, such as an external watchdog, in order to force a power-cycle of the device.

Hardening MultiProcessor System-On-Chips

MultiProcessor System-On-Chips (MPSoCs) are complex devices comprising multiple processing cores, as well as other processing units, memories and specialized hardware in the same chip instead of having those functions in multiple chips connected together. Technology scaling has allowed for the development and adoption of this kind of devices in multiple systems for ground applications, because of their high performance, low energy consumption and small size. MPSoCs are present in every modern smartphone and computer.

Alone for these characteristics, they could be attractive as COTS for space applications, but some of them also provide an interesting feature that can be taken advantage of to achieve a higher reliability in radiation environments. Devices such as AMD Xilinx' Zynq-7000 [138], Zynq Ultrascale+ [139] or Versal ACAP [140] include an FPGA as well as multicore microprocessors and other dedicated hardware for specific purposes like Digital Signal Processing, Artificial Intelligence, Radio Frequency communications or Video processing.

Several solutions to harden MPSoCs leveraging their special capabilities have been explored in recent years:

- **Trace interface**
  The trace interface is a means included in modern microprocessors to support debugging in the design phase. The trace provides online information about the state of the processor and the last instructions executed at any given time and it is useful to characterize the behaviour of the system under asynchronous events or exceptions. Once the application has been tested, this interface is often dismissed. However, the information it provides can be used to diagnose faults caused by SEE during normal operation. In [141] [142], the authors make use of the trace information to detect errors in the application or in the data flow, by detecting Program Counter (PC) addresses outside the expected range and data accesses outside user-specified ranges respectively. In [128], the authors propose a lockstep architecture able to detect errors during execution and perform rollbacks to retry the faulty operation, aided by a trace decoder capable of detecting program flow errors implemented in the programmable logic of a Zynq MPSoC.
- **Macro-synchronized Lockstep**
  Micro-synchronized lockstep requires the usage of ad-hoc designed processors, as

well as some support hardware to perform data exchange between the redundant processors. The Duplex approach proposed in [143] and [144] is a highly flexible and low-cost alternative to micro-synchronized lockstep implemented in a Zynq MPSoC. In this technique, two COTS processors execute the same application and their states and results are compared at check points distributed along the application flow, hence the macro-synchronization. The context of both cores is periodically stored in ECC-hardened Block RAMs implemented in the FPGA of the MPSoC. At check points, the cores exchange information also through the FPGA and upon error, the stored context from previous check points can be used to rollback the execution and retry the faulty operation. Thanks to the stored context, the application can be rolled-back to any point up to a configurable number of previous check points, allowing consecutive rollbacks when the immediately previous context is not able to produce correct results. Watchdogs and other supplementary hardening techniques are used to mitigate the effects of functional interrupts in the microprocessors. The basics of this approach are similar to those of the Dual Duplex Tolerant to Transients (DT2) architecture presented in [126], but the external ASIC used to provide macro-synchronization in the DT2 is implemented in the programmable logic of the MPSoC in this design.

2.7. Final considerations

As we have seen, the effects of radiation on electronics is a multifaceted and evolving research topic. The study of the basic mechanisms through which radiation is capable of damaging the newer technologies is challenging enough by itself. Furthermore, when seen under the engineering perspective, the problem of radiation can be tackled from a wide variety of approaches, adding extra complexity to the research topic, and, as a result, multiple solutions have been proposed over the years. Space systems, and other systems that may be subject to radiation in the next decades due to technology scaling, demand nowadays higher computational capabilities with a balanced power consumption, costs and development time. The research carried out in this Thesis and laid out in the next Chapters presents novel solutions to protect digital circuits against the effects of radiation. The proposed techniques rely on approximate computing to correct errors caused by radiation and reduce the power and resource consumption compared to traditional error mitigation techniques.

3. ERROR SENSITIVITY STUDY OF FFT ARCHITECTURES IMPLEMENTED IN FPGA

With the final objective of studying Approximate Error Mitigation techniques for digital circuits implemented in FPGA, we started this Thesis with a preliminary study on the behavior of the most commonly used error mitigation technique, the Triple Modular Redundancy. To test this hardening technique we chose to apply Block TMR to a typical Digital Signal Processing circuit, the Fast Fourier Transform, for which several configurations have been explored in the literature.

The experiments we present in this Chapter pursued two objectives. First, provide a baseline for future experiments regarding sensitivity and error classification of the BTMR technique applied to a relevant benchmark. These results helped us establish comparisons in terms of overhead and error correction capabilities between the BTMR and the Approximate Error Mitigation techniques we developed in this Thesis. And second, we used these experiments as a pretext to develop and consolidate the fault injection setup we used throughout the rest of this Thesis.

This chapter has been published as an article:

[2] © 2021 IEEE. Reprinted, with permission, from L. Garcia-Astudillo, A. Lindoso, L. Entrena, *et al.*, “Error sensitivity study of FFT architectures implemented in FPGA,” *Microelectronics Reliability*, vol. 126, p. 114 298, Nov. 2021. DOI: 10.1016/J.MICROREL.2021.114298

Abstract

This work studies the impact that the architectural choices can have in the error mitigation of a digital processing module such as the Fast Fourier Transform. To this purpose, several serial and pipelined architectures were implemented in FPGA using Block Triple Modular Redundancy and analysed under a fault injection approach that was previously validated with radiation. These architectures were compared with respect to error rate, common mode failure rate and signal-to-noise ratio. Experimental results show that the error rate is strongly correlated with the use of resources when using a similar architecture. However, pipelined architectures tend to have more common mode failures but with lower signal-to-noise ratio than a serial architecture.

3.1. Introduction

FPGAs (Field-Programmable Gate Arrays) are widely used today for Digital Signal Processing (DSP) applications. Since the introduction of embedded

multiplier-accumulator blocks, usually called DSP blocks, FPGAs can provide very high performance for DSP applications, exceeding that of state-of-the-art microprocessors and digital signal processors. Furthermore, the main advantage of FPGAs is the ability to tailor the implementation to match system requirements. FPGA inherent parallelism can be used to boost performance, while serial architectures can be used to save resources in low rate applications. As a matter of fact, DSP is a major application field for FPGAs.

FPGA-implemented DSP designs are increasingly used in safety and mission-critical applications, such as automotive, avionics and space. In this case, soft errors are an increasing concern and fault-tolerant designs must be used to mitigate them. For space, a few types of radiation-hardened FPGAs are available, but they are expensive and lag way behind their commercial counterparts. Therefore, designers are considering the use of COTS (Commercial Off-The-Shelf) FPGAs and mitigate errors by design. A typical solution is TMR (Triple Modular Redundancy) [94] combined with scrubbing to prevent error accumulation in the configuration memory. This solution is quite effective [145] . Nevertheless, it cannot mask all errors, because some configuration bits can cause Common Mode Failures (CMFs) [146].

In this work, we perform a study of soft error mitigation for FFT (Fast Fourier Transform) architectures. The FFT is a prime example of a DSP algorithm and it is widely used in a large variety of applications. Because it is a very important and complex algorithm, many architectures have been proposed to optimize performance and resource usage. The goal of this work is to evaluate some of the most representative architectures with respect to the reliability they can provide when implemented in an FPGA using TMR, and determine the impact that the choice of architecture has on the soft error resilience. To this purpose, we compare several parameters such as the error rate, the CMF rate and the Peak-Signal-to-Noise Ratio (PSNR).

In order to be able to compare a significant amount of different designs, the evaluation has been carried out by extensive fault injection campaigns. Nevertheless, radiation test results obtained with the same experimental setup show a very good correlation between radiation results and fault injection results with our approach. The results of this work demonstrate that although the error rate is primarily determined by the size of the design, the choice of architecture has a significant impact on the amount of unmitigated errors (CMFs) and the relevance of the errors. In particular, pipelined architectures tend to have more CMFs but with higher PSNR than a serial architecture.

The remaining of the paper is as follows. Section 3.2 summarizes related work. Section 3.3 describes the FFT architectures under study. Section 3.4 presents and discusses the experimental results. Finally, section 3.5 shows the conclusions of this work.

3.2. Related work

The problem of designing fault-tolerant circuits for DSP, and for FFT networks in particular, has traditionally received significant attention [147]-[148]. Early work focused on weighted checksum codes [147], [149], Concurrent Error Detection (CED) [149], [149], [150], [151] and time-redundant networks using redundant processing elements [152], [153], [148]. Some works proposed ad-hoc architectures that take advantage of the properties of the FFT algorithm, such as Algorithm-Based Fault Tolerance (ABFT) [154]. These ideas have been further developed in more recent works [102].

For the early approaches, the target technology was mainly ASIC (Application Specific Integrated Circuit) because FPGAs were hardly large enough to implement these complex algorithms. Performance and area overheads were of the utmost importance, and hence the solutions tried to provide fault-tolerance with much lower overhead than TMR. Moreover, these approaches were usually validated using simplified error models or theoretical estimations of error bounds. These models may not be appropriate for SRAM-based FPGAs, which have fine-grain configurable architectures that are affected by configuration memory errors.

Today, FFT designs for real-time applications are often implemented in FPGAs, taking advantage of the inherent parallelism of these devices. In this case, TMR can be a convenient solution that does not enforce a particular FFT implementation architecture. However, TMR is not fully effective because there are single configuration bits that can cause multiple errors across domains [146], usually known as Common Mode Failures [119]. For instance, a single configuration bit error can create a short or a double open, that may affect two of the three TMR domains or the voter circuit. F. L. Kastensmidt et al. [155] analyse the optimal placement of voters using a DSP case study to reduce CMFs. Another source of CMFs are Multiple Cell Upsets (MCUs), which are multiple bit errors caused by the strike of a single particle. An MCU can create a CMF if the multiple upsets are not in the same TMR domain. Nevertheless, although MCUs are becoming relevant for advanced technologies, they are still much less frequent than SEUs [119]. Thus, SEUs are still the cause of most CMFs in SRAM-based FPGAs [123].

3.3. FFT architectures under study

FFT architectures can be divided in two main groups: serial and pipelined. They can be selected according to the application requirements. Serial architectures offer resource efficiency, while pipelined architectures aim at higher throughput. In particular, pipelined architectures can support the continuous processing of input data which is needed in real-time processing applications.

Serial FFT

Serial architectures perform the required mathematical operations in an iterative way over the same circuit, storing the intermediate results in banks of memory. Internally, the circuit can be designed in various ways, of which the most popular are the Radix-2 and Radix-4 algorithms. They differ in how the input data are split in groups and passed to the butterfly unit (BU) that performs complex additions. Butterflies in Radix-2 perform 2 complex additions, while those in Radix-4 execute 4 additions. After the butterflies, the data are multiplied by the twiddle factors, which are complex constant numbers related to the length of the transform. The Radix-4 algorithm is considered more efficient, as it divides the number of iterations by 2 with respect to the Radix-2 algorithm. In this work, we will study the performance of a Radix-4 serial circuit, implemented using Xilinx' configurable FFT IP.

Pipelined FFT

Pipelined versions of the FFT algorithm are based on chains of stages, comprising Butterfly Units (BUs) and complex multipliers. The output of each stage feeds the next stage until all input data have been processed. To feed the BUs with the correct data, it is necessary to add data delays between stages. The Single- and Multi-path Delay Commutator (SDC and MDC, respectively) use memory blocks and multiplexors to create these buffers [156]. The Single-path Delay Feedback (SDF) approach [157] introduces FIFO memories at the output of the BUs and feeds their contents back to the BUs when necessary using control signals. SDF designs are generally more efficient in the usage of memory and we selected this approach for our designs.

As in the case of the serial FFT, various radix algorithms can be used. In this work we explored the features of Radix-2, Radix-4 and Radix-2^2. The Radix-2 architecture (R2SDF), depicted in Fig. 3.1a, is the simplest of the three. It encompasses log2(N) stages, where N is the transform length. Each stage has a 2-input BU, a FIFO memory to implement data delays and a complex multiplier.

Fig. 3.1b represents a Radix-4 architecture (R4SDF). In this case, log4(N) stages are needed. The stages enclose a 4-input BU, 3 FIFO memories and one complex multiplier. The FIFOs are all of the same size within the stage, but their length is reduced as the data advances through the pipeline.

The Radix-2^2 (R22SDF) architecture, shown in Fig. 3.1c, also contains log4(N) stages. However, each stage contains two slightly different BUs, a complex multiplier and two FIFO memories, whose depth depends on the order of the stage. In the first stage the FIFOs are N/2 and N/4 long, and then they are halved in each stage. The Radix-2^2 approach combines features of the Radix-2 and Radix-4 architectures, and it is more efficient in the usage of resources than both of them.

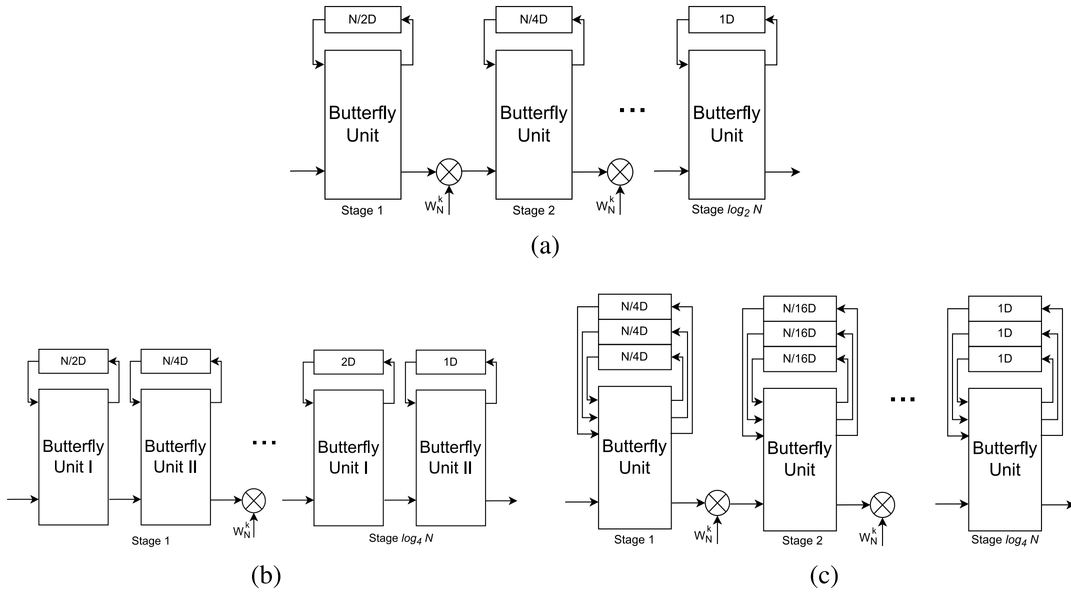


Figure 3.1: Pipelined FFT architectures: (a) R2SDF, (b) R4SDF, and (c) R22SDF.

3.4. Experimental results

To assess the performance of the proposed architectures, we developed 64 and 256 points versions of the FFTs and implemented fault-tolerant versions using a Block TMR approach. Each of them was wrapped on a testbench circuit capable of providing inputs for the FFTs, comparing the outputs against the expected results and collecting error data for off-line analysis. Upon error, the results from each instance of the FFT and their voted output are sent through a UART connection to the external host that controls the experiments.

The programmable logic of a Xilinx Zynq-7010 All Programmable SoC (APSoC) was used in these experiments to implement the designs. The dual core ARM processor included in this device was not used in the tests. We used a default effort in the synthesis and implementation of all designs, with a common clock constraint. We have just used constraints to ensure redundant FFT instances were not optimized away.

Fault injection was implemented using Xilinx Soft Error Mitigation (SEM) IP [74]. This IP can be used to detect or correct errors in the configuration memory, and also to inject faults. Faults were injected randomly at a sufficiently slow rate to ensure that a complete FFT can be computed between faults. This setup can also be used for radiation test experiments, by simply disabling fault injection. We performed one radiation test experiment at Centro Nacional de Aceleradores (CNA) in Seville with low energy protons (up to 15 MeV) using the 64-point serial TMR design and the results matched very well with fault injection. Specifically, the ratio of CMFs to the total errors was 1.29% under radiation and 1.04% under fault injection. However, we used fault injection for the analysis because the amount of beam time required to test all the versions would have been prohibitive.

Table 3.1 reports the hardware utilized by each architecture for the 64-point and 256-point FFT versions. It is evident from the data in the table that the Radix-22 architecture is the most efficient of the pipelined versions regardless the transform length, while Radix-2 and Radix-4 are similar in terms of total area used. The Serial Radix-4 consumes far more resources in the 64-point FFT, but it is with longer FFTs where this architecture shows its efficiency. The addition of more stages in the pipeline greatly increases the resources necessary in the R2, R4 and R22 architectures, whereas the serial FFT is barely affected by the change.

Table 3.1: RESOURCE UTILIZATION OF THE DIFFERENT ARCHITECTURES

	R2SDF FFT		**R4SDF FFT**		**R2^{2}SDF FFT**		**Serial R4 FFT**	
Points	64	256	64	256	64	256	64	256
Stages	6	8	3	4	3	4	1	1
LUT as logic	1065	1627	1231	1939	971	1445	1072	1106
LUT RAM	218	802	236	821	221	805	181	188
FF	1069	1437	705	930	723	930	2134	2227
BRAM	0	0.5	0	0	0	0	3.5	3.5
DSP Slices	15	21	6	9	6	9	9	9

Table 3.2 summarizes the fault injection results. The first row shows the number of injections performed in each design. The next two rows show the total amount of erroneous FFT calculations (faulty frames) and the error rate, respectively. The error rate is calculated as the ratio of faulty frames to the number of injected faults. We define a word as the real or imaginary part of each complex point in an FFT frame. This way, a frame of length N, with N being 64 or 256 points in this study, would have 2N words. The results of the three FFTs and the voted output are collected when an error occurs, and the words of each frame can be later compared with correct results. This way, the comparison can establish whether errors affected one FFT, several FFTs or the voting logic, and we can classify the errors according to this information. In case an error affects a word in just one FFT, the voter mitigates the error and it will be classified as a masked error. On the contrary, if the fault affects more than one of the FFTs, the triplicated voters, or the voting logic and at least one of the FFT instances, the voter is not capable of correcting the fault and the error will be classified as an unmitigated error or Common Mode Failure (CMF). CMFs are reported in the fourth row of Table 3.2. Finally, the last row in Table 3.2 reports the CMF rate, which is calculated as the ratio of CMF frames to the number of injected faults. The error rate and the CMF rate of the designs are shown in Fig. 3.2 for comparison.

Table 3.2: RESULTS OF THE FAULT INJECTION CAMPAIGNS

	R2SDF FFT TMR		R4SDF FFT TMR		R2^{2}SDF FFT TMR		Serial R4 FFT TMR	
Points	64	256	64	256	64	256	64	256
Number of injections	216000	357120	216000	367200	216000	422640	216000	388800
Faulty frames (total)	812	1649	886	1775	748	1861	816	1416
Error Rate (10^{-3})	3.76	4.61	4.09	4.83	3.46	4.40	3.78	3.64
CMF frames	30	51	24	33	26	48	10	19
CMF Rate (10^{-4})	1.28	1.43	1.11	0.89	1.20	1.14	0.46	0.49

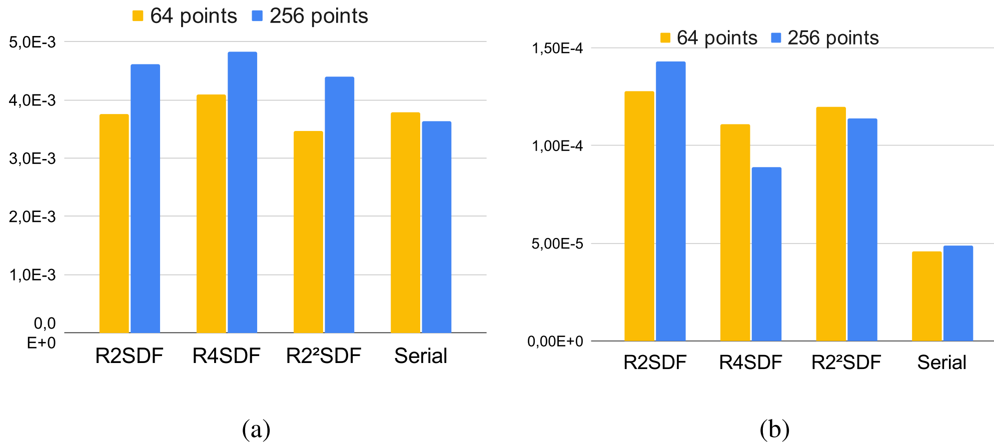


Figure 3.2: Error rates (a) and CMF rates (b) of the tested architectures.

By carefully analysing the error rates in Fig. 3.2a, we can clearly see that all the 64-point designs behave in a similar way, but when the FFT is enlarged to 256 points, only the pipelined designs increase their error rate. This suggests a strong correlation of the error rate with the resources used by the architecture, as shown in Fig. 3.3.

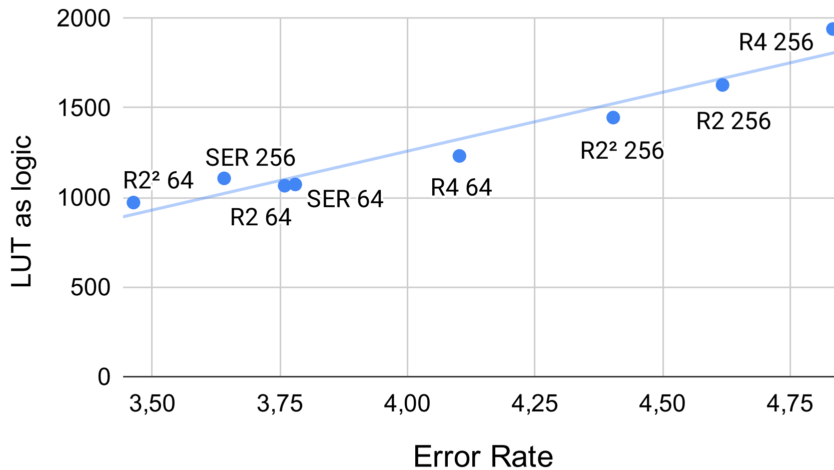


Figure 3.3: LUT as logic utilization vs. error rate.

A similar correlation can be established between the CMF rate and the resources.

These two correlations seem to suggest that a higher utilization of resources leads to a higher incidence of faults, which, in turn, increases the appearance of CMFs. However, as seen in Fig. 3.2b, the CMF rate is between 2 and 3 times smaller in the serial case than in the pipelined designs, even in cases where the difference in area favours the pipeline architecture, such as the 64-point R22SDF architecture. Thus, the architecture has a significant impact on the criticality of the errors. This is also noticeable when comparing the pipeline designs: although the R4 architecture is significantly larger than the others, its CMF rate is lower than the CMF rate of R2 and R22 architectures.

A more detailed analysis is shown in Table 3.3, which reports the total amount of faulty words in the faulty frames. What stands out in Table 3.3 is a lower incidence of erroneous words in pipelined designs than in the serial architecture, which is even more evident when the transform length is increased to 256 points. Contrary to expectations, the data obtained suggest that a longer pipeline has a lower percentage of its outputs affected by faults. This result may be explained by the separation of computations in stages and the truncation performed between the stages to avoid overflow. An error affecting the least significant bits would be mitigated by this truncation. Thus, the more stages are in the pipeline, the less likely error propagation is.

This effect has another positive consequence: the percentage of words affected by CMFs is also lowered with longer pipelined FFTs. However, the serial implementation does not exhibit this kind of behaviour. Because all the stages are performed by the same hardware, a configuration memory error in the butterfly or the multiplier would affect the result of every stage, while in a pipeline, only one result would be corrupted, and chances are higher that it may be mitigated in the following stages.

A graphical representation of these two phenomena is provided in Fig. 3.4. This figure shows the error classification of all the words in the faulty frames found, expressed as a percentage of the total. The percentage of correct words is significantly higher in longer pipeline FFTs, and the percentage of words with CMFs is also reduced.

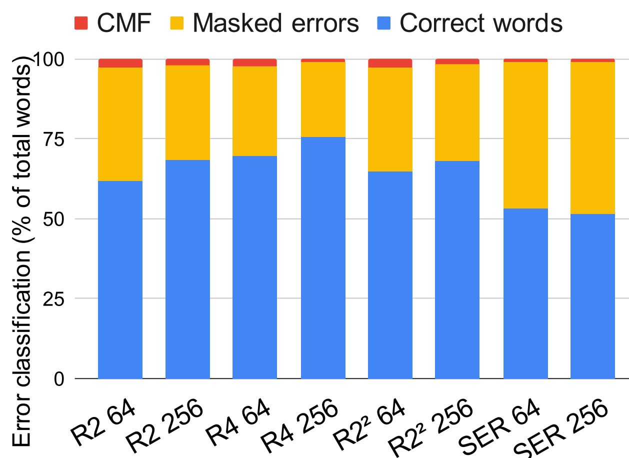


Figure 3.4: Error classification of the words in faulty frames.

The last row of Table 3.3 sheds additional light over this finding. This row presents the average Peak Signal-To-Noise Ratio (PSNR) of all the faulty frames, a measure of the distortion of a signal due to noise caused by errors. Eq. 3.1 describes the method to compute the PSNR of a faulty frame, where MAX is the highest possible value of the signal and the MSE is the Mean Squared Error of the FFT frame, calculated as the sum of the squared error of each point in the frame divided by the number of points in the frame.

$$PSNR = 10 \times log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{3.1}$$

From the data in Table 3.3, we can see that the PSNR improves significantly in the longer pipelined designs. Not only the number of total errors is decreased due to truncation, but also the importance of the errors is lowered. All pipeline designs have a better performance in terms of PSNR than the serial architecture. Although seemingly contradictory with the previous results presented, this finding is, in fact, very interesting. The experiments hint that pipelined FFT designs are more sensitive to radiation effects due to their larger area overhead, but at the same time, they boast of an intrinsic error mitigation mechanism that allows them to eliminate small errors and lower the impact of more significant errors.

A comparison between the pipeline designs can also be carried out in these terms. As in the case of the CMF rate, the R4 architecture seems to perform better, while the R2 and R22 architectures are similar. The percentage of correct words per frame in the R4 architecture is the highest of the three pipeline designs and, accordingly, the PSNR of the R4 architecture is also outstanding.

Table 3.3: ERROR CLASSIFICATION OF THE REPORTED FAULTS

	R2SDF FFT TMR		**R4SDF FFT TMR**		**R2^{2}SDF FFT TMR**		**Serial R4 FFT TMR**	
Points	64	256	64	256	64	256	64	256
Total words in faulty frames	103936	844288	113408	908800	95744	952832	104448	724992
Faulty words	39751	267648	34444	220484	33572	305808	48854	350785
(real or imaginary)	(38.2%)	(31.7%)	(30.37%)	(24.3%)	(35.1%)	(32.1%)	(46.7%)	(48.4%)
Masked errors (%)	92.6	93.6	92.9	95.8	92.2	94.6	98.1	98.2
Words with CMFs (%)	7.4	6.4	7.1	4.2	7.8	5.4	1.9	1.8
PSNR (dB)	70.1	77.3	78.9	82.1	72.6	78.8	69.5	72.9

3.5. Conclusions

In this work we performed a comparison of the error sensitivity of several FFT architectures implemented in a SRAM-based FPGA using fault injection. Our results showed a clear relationship between the usage of resources and the sensitivity to Single Event Upsets, giving an evident advantage to serial architectures, that only increase their size slightly when computing larger FFTs, whereas the addition of more stages in the pipeline versions imposes a penalty on their sensitivity. However, we also showed that the architecture has a significant impact on the criticality of the errors. Our data suggest

that the incidence of critical errors (Common-Mode Failures) in serial implementations is much lower than in pipeline architectures, even in cases where the pipeline architecture has a smaller area. There are also significant differences among the pipeline architectures that do not correlate with the size of the implementation.

Nevertheless, regarding the quality of the results, the reutilization of resources in a serial architecture leads to a higher number of erroneous data points in the FFT frame and noisier results due to error accumulation in the stages.

Acknowledgments

This work has been supported in part by the Spanish Ministry of Science and Innovation under project PID2019-106455GB-C21 and by the Community of Madrid under project no. 49.520608.9.18.

References

[94] M. Berg, "Single Event Effects in FPGA Devices 2014-2015", NASA Electronic Parts and Packaging Program (NEPP) Electr. Tech. Workshop (ETW), June 2015.

[145] K. S. Morgan, D. L. McMurtrey, B. H. Pratt, and M. J. Wirthlin, “A comparison of TMR with alternative fault-tolerant design techniques for FPGAs”, IEEE Transactions on Nuclear Science, vol. 54, no. 6, pp. 2065–2072, Dec. 2007.

[146] H. Quinn et al. “Domain crossing errors: Limitations on single device triple-modular redundancy circuits in Xilinx FPGAs”, IEEE Transactions on Nuclear Science, vol. 54, no. 6, pp. 2037–2043, Dec. 2007.

[147] J.-Y. Jou and J. A. Abraham, "Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures", Proceedings of the IEEE, vol. 74, no. 5, pp. 732-741, May 1986.

[149] J.-Y. Jou and J. A. Abraham, "Fault-tolerant FFT networks", IEEE Transactions on Computers, vol. 37, no. 5, pp. 548-561, 1988.

[158] D. L. Tao and C. R. P. Hartmann, "A novel concurrent error detection scheme for FFT networks", IEEE Transactions on Parallel and Distributed Systems, vol. 4, no. 2, pp. 198-221, 1993.

[159] C. G. Oh and H. Y. Youn, "On concurrent error location and correction of FFT networks", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 2, no. 2, pp. 257-260, June 1994.

[150] F. Lombardi and J. C. Muzio, "Concurrent error detection and fault location in an FIT architecture", IEEE J. Solid- State Circuits, vol. 27, no. 5, pp. 728-736, 1992.

[151] J.-F. Li, S.-K. Lu, S.-A. Hwang and C.-W. Wu, "Easily testable and fault-tolerant

FFT butterfly networks", IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, no. 9, pp. 919-929, Sept. 2000.

[152] A. Antola, R. Negrini, M. G. Sami and N. Scarabottolo, "Fault-tolerance in FFT arrays: time-redundancy approaches", IEEE Int. Conf. on Communications, vol.3, pp. 779-785, 1990.

[153] Y.-M. Hsu and E. E. Swartzlander, "FFT arrays with built-in error correction", Proc. 28th Asilomar Conf. on Signals, Systems and Computers, vol.1, pp. 172-176, 1994.

[148] T.-H. Chen and L.-G. Chen, "Concurrent error-detectable butterfly chip for real-time FFT processing through time redundancy", IEEE J. Solid-state Circuits, vol. 28, no. 5, pp. 537-547, 1993.

[154] K.-H. Huang and J. A. Abraham, "Algorithm-Based Fault Tolerance for Matrix Operations", IEEE Transactions on Computers, vol. C-33, no. 6, pp. 518-528, June 1984.

[102] P. Reviriego, C. Bleakley, J. A. Maestro and A. O'Donnell, "Offset DMR: A Low Overhead Soft Error Detection and Correction Technique for Transform-Based Convolution", IEEE Transactions on Computers, vol. 60, no. 10, pp. 1511-1516, Oct. 2011.

[119] M. Wirthlin, D. Lee, G. Swift, and H. Quinn, "A method and case study on identifying physically adjacent multiple-cell upsets using 28-nm, interleaved and SECDED-protected arrays", IEEE Transactions on Nuclear Science, vol. 61, no. 6, pp. 3080–3087, Dec. 2014.

[155] F. L. Kastensmidt, L. Sterpone, L. Carro and M. Sonza Reorda, "On the Optimal Design of Triple Modular Redundancy Logic for SRAM-Based FPGAs", Proc. Design, Automation and Test in Europe, vol. 2, pp. 1290–1295, 2005.

[123] M. J. Cannon, A. M. Keller, H. C. Rowberry, C. A. Thurlow, A. Pérez-Celis and M. J. Wirthlin, "Strategies for Removing Common Mode Failures From TMR Designs Deployed on SRAM FPGAs", IEEE Transactions on Nuclear Science, vol. 66, no. 1, pp. 207-215, Jan. 2019.

[156] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation", Proc. URSI Int. Symp. on Signals, Systems, and Electronics, pp. 257-262, 1998.

[157] E.H. Wold and A.M. Despain. "Pipeline and parallel-pipeline FFT processors for VLSI implementation". IEEE Trans. Comput., C-33(5), pp. 414-426, May 1984.

[74] "Soft error mitigation controller v4.1 Product guide", Xilinx Inc., White Paper PG036, Nov. 2014.

4. ANALYZING REDUCED PRECISION REDUNDANCY UNDER PROTON IRRADIATION

Having performed an exhaustive analysis on the behavior and characteristics of the BTMR technique under fault injection, we developed Reduced Precision Redundancy benchmarks to be tested under fault injection and also under proton irradiation. For these experiments we used yet another architecture of the Fast Fourier Transform benchmark, trying different configurations to test their influence on the error mitigation.

The irradiation campaign allowed us to validate that the setup we developed for the fault injection experiments could be used for irradiation with minor modifications. By using the same setup we were able to correlate the fault injection and irradiation results.

This chapter has been published as an article:

[1] © 2022 IEEE. Reprinted, with permission, from L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, *et al.*, "Analyzing Reduced Precision Triple Modular Redundancy under Proton Irradiation," *IEEE Transactions on Nuclear Science*, vol. 69, pp. 470–477, 3 Mar. 2022. DOI: 10.1109/TNS.2022.3152088

Abstract

This work analyzes the performance of the Reduced Precision Redundancy (RPR) error mitigation technique using the Fast Fourier Transform (FFT) as a case study. To this purpose, several configurations of an FFT IP design were implemented in FPGA using Reduced Precision TMR and tested under proton irradiation and fault injection. The cross-section, the sensitivity to Common-Mode Failures (CMFs) and the signal-to-noise ratio of these configurations were evaluated. The results of the radiation experiments and fault injection campaigns are in agreement and show that the Reduced Precision TMR technique may be used as an alternative to TMR if small errors can be tolerated, as it has a good performance in terms of error correction capabilities, area usage and sensitivity to critical errors.

4.1. Introduction

FPGAs can provide high performance and flexibility for the implementation of digital circuits. For this reason, they are becoming a key component in a large variety of applications, including space and safety-critical ones. For space, there are radiation hardened FPGAs which can meet radiation requirements. However, the capabilities of radiation-hardened FPGAs are far below those of commercial devices. For low-cost satellites or less critical parts of some missions, commercial FPGAs are being considered

as a means to increase on-board computational power and reduce size and power consumption.

When unhardened technologies are used, error mitigation must be implemented by design. Triple Modular Redundancy (TMR) is a well-known and widely used error mitigation approach. However, it introduces a high overhead. To alleviate this problem, Approximate TMR (ATMR) techniques have been proposed [160], [161], [162]. ATMR consists in using approximate copies instead of full redundant copies of the design. The goal is to reduce the overhead while keeping as much as possible the capability to correct errors. Reduced Precision Redundancy (RPR) [97], [98] is an approximation approach that consists in using reduced precision data and operations. The rationale behind RPR is that a reduced precision module should produce a less precise but similar output, so that large errors can be detected or masked by comparing the outputs of different modules. A Reduced Precision TMR (RP-TMR) implementation can be used to obtain a correct result, or at least an acceptable approximate correct result, to the benefit of reduced overhead.

Although RPR was proposed some years ago [4-7], existing approaches have been mainly evaluated through fault injection. To the best of our knowledge, we have not found any former work reporting radiation results on an RPR design. In this work, we perform an evaluation of RPR under low energy proton irradiation. As a case study, we selected a Fast Fourier Transform (FFT) IP. The FFT is a representative example of Digital Signal Processing that requires a significant amount of storage and arithmetic resources. The evaluation of an RP-TMR FFT IP is also novel, as former RPR studies were generally based on FIR filters. We selected a library IP design of the FFT benchmark to illustrate a worst-case example, for which designers have limited control over the architecture.

Finally, a very important goal of this work is to evaluate the sensitivity of RP-TMR to critical errors, i.e., to errors that are not masked. In particular, it has been shown that TMR designs in FPGAs are susceptible to Common Mode Failures (CMFs), which are errors that cause more than one copy to fail simultaneously. CMFs are difficult to avoid because there are single bits in the FPGA configuration memory that can have a multiple effect [122], [146], [123]. Previous work has shown that ATMR can produce a shift of the critical cross-section to the non-critical cross-section with respect to the TMR circuit [163].

In this work we show that RPR is less sensitive to faults and can also reduce the appearance of CMFs with respect to TMR. The results of the radiation experiments and fault injection campaigns we performed correlate quite well and prove that the RPR technique has a good performance in terms of error correction capabilities, area usage and sensitivity to critical errors.

The rest of this paper is organized as follows. Section 4.2 reviews the literature on the topic. Section 4.3 describes the Reduced Precision Redundancy mitigation technique and how we applied it to a variety of FFT benchmark versions in order to evaluate its capabilities. Section 4.4 covers the experimental setup and the facilities used for the

experiments. Section 4.5 presents the results of the experiments. Finally, Section 4.6 presents the conclusions of this work.

4.2. Related work

In many intensive computing applications, the idea of approximate computing is becoming appealing to improve efficiency. Approximate computing techniques [164] relax accuracy requirements to reduce complexity, improve energy efficiency and increase performance. The use of approximate computing to implement efficient fault tolerant architectures is discussed in [165]. In Digital Signal Processing (DSP), approximation is commonly used in many ways, such as sampling, quantization, or decimation. In fact, it can be considered that DSP algorithms are generally approximate, with a degree of approximation that depends on the parameters chosen for each particular application. Thus, there are always some sources of inaccuracy, both internal and external, which are commonly referred to as “noise”.

On the contrary, error correction techniques, such as TMR, are commonly designed to produce exact results, using exact redundant copies of the design at the expense of a high overhead. In the case of SRAM-based FPGAs, the entire design must be made redundant because errors in the configuration memory can affect the combinational and the sequential components, as well as the routing [94]. Exact redundancy has a large impact on area and power consumption. Therefore, it makes sense to reduce the accuracy of the redundant modules so as to accept errors that are within tolerable levels of noise.

Approximate TMR techniques relax the replication accuracy to reduce the hardware overhead. As a side effect, a smaller area contributes to reduce the radiation-induced error sensitivity. In [160] and [161], approximation techniques are proposed for logical circuits. They use very general, fine-grain approaches that are based on approximating logic functions. In [163], this idea was extended to FPGA designs by considering the approximation of the logic functions implemented by Look-Up Tables. The approach was validated with protons [163] and afterwards with neutrons [162]. An interesting result of these works is that although some errors are not corrected, the critical errors are reduced. ATMR reduces the number of critical configuration bits and can even be more robust than full TMR, especially when faults are allowed to accumulate in the FPGA configuration memory [162].

In a recent work [166], Keller et al. explored partial TMR as a means to reduce the overhead. They selectively replicate some components, but not others, and evaluate several selection approaches under neutron radiation and fault injection. This study shows a wide variety of results, where some selections can improve the cross-section while others can do even worse than the unmitigated design.

The idea of approximate TMR has also been applied to software running in a processor. In [167], the authors focus on successive approximation algorithms and

analyze the trade-off between fault masking and execution time by varying the number of iterations.

For DSP designs, approximations can be based on the parameters of the computation. Reduced Precision Redundancy is a systematic approximation approach, originally proposed by Shim et al. [97], [98], that consists in using replicas of the original circuit with reduced precision operands. It was intended to enable power consumption reduction by voltage overscaling and compensate for the soft errors introduced when voltage is lower than the critical voltage required for correct operation. This idea was extended in [168], with an in-depth analysis of two major parameters of the approach, namely the bit width and the approximation threshold. They studied the selection of the approximation threshold using theoretical and experimental approaches. To this purpose, the authors performed fault injection campaigns on a set of FIR filter designs with several approximation thresholds. Then, they also studied the bit error rate for different bit widths. All these works use digital filters as a case study and evaluated the designs using fault injection. The reliability of reduced precision matrix multiplication implemented with High-Level Synthesis (HLS) tools was analyzed in a recent study [169].

For the FFT algorithm, preliminary work using fault injection was presented in [8]. The results hinted beneficial side-effects of reducing the usage of resources through RPR. RP-TMR showed a lower error rate than a conventional TMR implementation, which, in turn, reduced the occurrence of Common-Mode Failures [8].

In [170], Liu et al. extended RPR to the case of N Modular Redundancy (NMR). They proposed a design for the decision logic and illustrate it for the case of 5 redundant copies. The evaluation was mainly theoretical, based on a probabilistic model that was proposed for this purpose.

4.3. Reduced Precision Redundancy

Error mitigation in FPGAs has traditionally been implemented by using Triple Modular Redundancy. SRAM-based FPGAs typically require Distributed TMR (DTMR) or Block TMR (BTMR) approaches to cope with errors in the configuration memory [94]. These techniques involve triplicating the entire circuit and hence they have a high impact on the power consumption and the resources used by the system. The Reduced Precision Redundancy technique pursues a trade-off between the resources needed to mitigate errors and the precision of the results affected by error.

The Reduced Precision TMR technique is based on the addition of two approximate copies to the design to protect. These copies utilize reduced precision data to produce a similar result to the one expected from the Full Precision (FP) instance. The two Reduced Precision (RP) and the Full Precision instances are assembled in a BTMR architecture. A majority voter like the one typically used in TMR cannot be used here because the results of the FP and the RP instances may be slightly different. However, by means of a smart

voting logic, errors affecting the Full Precision result can be corrected to some extent by using the Reduced Precision results. Fig. 4.1 illustrates the architecture of the RP-TMR voter, applied to the hardening of an FFT module.

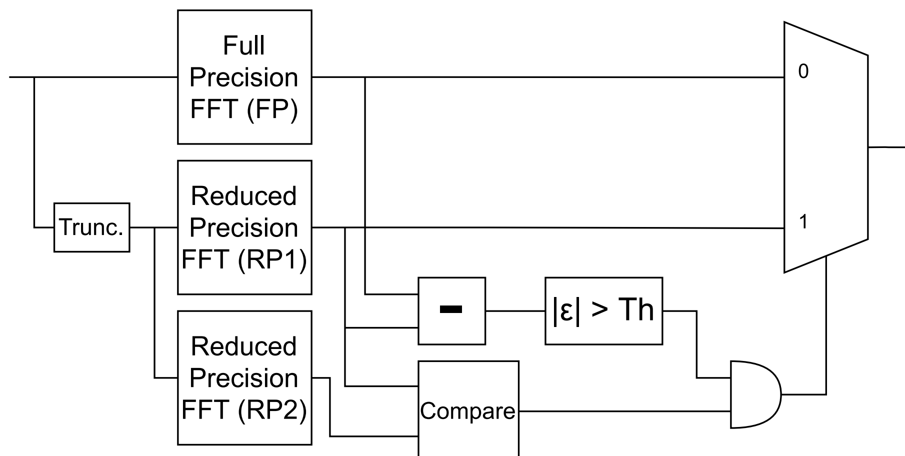


Figure 4.1: Architecture of an RP-TMR design to harden an FFT.

The voting process works in two phases. First, the FP and the RP results are compared to detect errors. A difference between them that is smaller than a predefined threshold value indicates a probably correct answer (an error smaller than the threshold will not be detected, but it is considered as a tolerable error). Should the difference be greater than the threshold, the second phase is initiated. In that case, if both RP copies yield the same result, the RP result is chosen as the final output of the voter, since it is considered more accurate than the result of the erroneous FP instance. This voting process is particularly correct under a single error assumption. If a single error occurs, it should only affect one of the modules. Thus, if the comparison between RP1 and RP2 fails, then the FP result should be correct. On the other hand, if RP1 and RP2 produce the same result, it should be a correct, but less accurate result. This way, the RP-TMR construct is capable of partially correcting wrong results, while keeping a level of noise at the output below the threshold value.

The main objective of the RP-TMR technique is the reduction of area and power needed by a full TMR design. The RP-TMR voting logic is more complex than the majority voter of a TMR implementation. However, the reduction achieved by decreasing the precision of the redundant copies is enough to significantly impact the utilization of resources. In turn, the area reduction can decrease the radiation sensitivity of the design.

Aiming to make an extensive study on RPR applied to the FFT algorithm, we developed and tested the benchmarks listed in Table 4.1. These benchmarks allow us to evaluate the impact of the FFT parameterization on the error mitigation, the cross-section, and the error distribution. Some basic FFT characteristics are shared among all the benchmarks: they implement 64 points calculations based on the Radix-4 algorithm working on serial mode ("Burst mode" in Xilinx terminology) using the Xilinx Fast Fourier Transform IP [171]. Other parameters, such as the word-length of the real and

imaginary parts of the input and output data or the resources used to implement the logical operations, are varied in the different benchmarks. Conventional full TMR (TMR 32-32-32) and unprotected FFT versions were also implemented to establish a comparison with RPR design. For the sake of clarity, the names of the designs include the sizes of the three instances. The sizes refer to complex data. Thus, for instance, a size of 32 bits is divided in 16 bits for the real part and 16 bits for the imaginary part.

Table 4.1: EXPERIMENTS PERFORMED AND THEIR FEATURES

Design	**CLB/DSP**	**FP size**	**RP size**
Unprotected FFT	CLB	32	-
TMR 32-32-32	CLB	32	-
RPR 32-16-16	CLB	32	16
RPR DSP 32-16-16	DSP	32	16
RPR 64-16-16	CLB	64	16
RPR 64-32-32	CLB	64	32
RPR 64-48-48	CLB	64	48

The RPR voting logic for an FFT compares separately the real and imaginary parts of the output data and that is the minimum correctable piece of data of our designs. The threshold in these voters has been chosen as the smallest one that does not trigger false positives in the error detection for each benchmark. This ensures low noise in the output when a fault affects the RP FFT. The threshold values guarantee maximum errors in the output that range from 3% in the case of RPR 32-16-16 to 3×10^{-6} in the case of RPR 64-48-48. In practice, the maximum tolerable error depends on the application and the sizes are selected so as to satisfy it. Fig. 4.2 shows the synthesis results for some of the designs listed in Table 4.1. These results evidence the advantage of RPR in the FFT application studied in this work. When comparing 32-bit FFTs built with Configurable Logic Blocks (CLB) using FP copies (TMR 32-32-32) and RP copies (RPR 32-16-16), we can observe 16% decrease in the number of LUTs, and 25% less flip-flops and LUTs used as RAM memory (LUTRAMs). Interestingly, doubling the precision of all the FFTs in an RPR (RPR 32-16-16 vs. RPR 64-32-32) does not necessarily incur in a double overhead penalty. Therefore, the difference in resources should be more evident with higher precisions. By implementing some of the mathematical operations in the FFT using DSP blocks instead of CLBs, the usage of LUTs can be further decreased at the cost of using more DSP blocks. All the TMR and RPR designs use triplicated voters to reduce the incidence of voting logic errors.

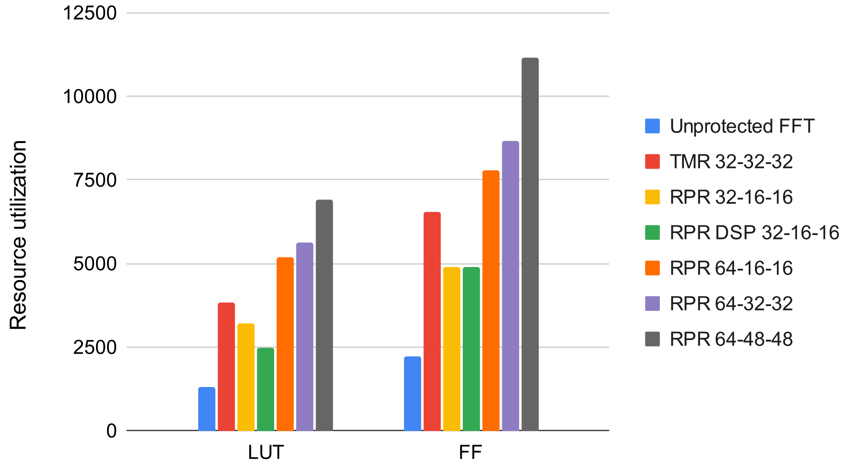


Figure 4.2: Utilization of resources (LUTs and FFs) of the tested benchmarks.

4.4. Experimental setup

To assess the radiation performance of RP-TMR and explore its capabilities, we implemented the benchmarks listed in Table 4.1 using Xilinx FFT IP and configuring some of the FFT parameters. The benchmarks have comparable performances, considering they all run at the same clock frequency and execute the calculations in the same number of clock cycles. They were also implemented using default placement strategies and a single clock domain for the three FFTs.

The tested designs were wrapped in a testbench circuit able to provide input stimuli, check the output results and, whenever an error is detected, send them through serial communication to an external host that is controlling the experiment. Upon error, the external host collects the output data from the three FFT copies and the voted result, which allow us to compare them against correct results to classify the errors and analyze their impact on the output.

All designs were implemented in the programmable logic of a Xilinx Zynq-7010 All Programmable SoC (APSoC) device [138]. Although this device contains a dual core ARM processor, it was not used in the experiments.

The designs were tested under a proton beam at Centro Nacional de Aceleradores (CNA) in Sevilla (Spain) in January 2021 using low-energy protons (15 MeV). The device has proven to be sensitive enough for this energy without thinning it in several previous experiments [172].

For the purpose of a comparative analysis, fault injection tests were also carried out besides the irradiation experiments. The Xilinx Soft Error Mitigation (SEM) IP was employed to inject faults in the configuration memory of the FPGA and investigate the impact of the errors on the implemented designs [74]. The setup used for fault injection was the same as that used for radiation testing.

4.5. Experimental results

This section covers the main findings resulting from the irradiation and injection experiments performed. Statistical analyses regarding error rates and error classifications are summarized to compare the effectiveness of the mitigation techniques.

Radiation test results

Table 4.2 compares the irradiation results in terms of the number of erroneous FFT calculations (frames) found in the tested designs. The first row of this table presents the total amount of faulty frames found. This quantity is decomposed in three possible error categories in the following rows:

- Masked errors: they occur in the RP modules and are masked by the voter, which selects the FP result in case the RP modules do not produce the same result.
- Tolerable errors: they occur in the FP module and produce a small variation with respect to the correct result. The voter guarantees that the error is below the selected threshold.
- Uncorrectable errors: errors that occur in more than one module or in the voter.

Table 4.2: RESULTS OF RADIATION EXPERIMENTS

	Unprotected 32-bit FFT	**TMR 32-32-32**	**RPR 32-16-16**	**RPR DSP 32-16-16**	**RPR 64-16-16**	**RPR 64-32-32**	**RPR 64-48-48**
Faulty frames (total)	216	386	241	199	214	229	304
Masked errors	-	382 (98.96%)	133 (55.19%)	111 (55.78%)	77 (35.98%)	113 (49.34%)	171 (56.25%)
Tolerable errors	-	-	105 (43.57%)	86 (43.22%)	136 (63.55%)	111 (48.47%)	132 (43.42%)
Uncorrectable errors	216 (100%)	4 (1.04%)	3 (1.24%)	2 (1%)	1 (0.47%)	5 (2.19%)	1 (0.32%)
Cross-Section (10^{-10} cm^2)	0.99 (0.86, 1.12)	1.82 (1.64, 2.01)	1.49 (1.3, 1.68)	1.83 (1.58, 2.09)	2.1 (1.82, 2.38)	2.64 (2.3, 2.98)	2.88 (2.56, 3.21)
CMF Cross-Section (10^{-12} cm^2)	-	1.89 (0.52, 4.84)	1.86 (0.38, 5.43)	1.84 (0.22, 6.65)	0.98 (0.02, 5.46)	5.76 (1.87, 13.4)	0.95 (0.02, 5.28)
PSNR (dB)	66.40	64.17	74.80	70.25	124.87	131.57	149.39

Closer inspection of the error classification results reveals a high number of tolerable errors in the output of the RPR systems, around 45% of the faulty FFT frames. This category includes frames whose errors were corrected with the approximate results from the RP FFTs and those whose errors were smaller than the threshold, which can be tolerated. Tolerable errors can only be found in RPR designs. While TMR effectively masks errors affecting one of the FFTs, RPR masks errors found in a Reduced Precision instance or large errors in the Full Precision instance. Faults in the Full Precision instance produce tolerable errors. Errors affecting either the voting logic or more than one FFT at the same time (Common-Mode Failures) cannot be corrected by these hardening techniques and minimizing their appearance is therefore of utter importance. Frames in which these errors appear are classified as uncorrectable errors in the table. The irradiation

and injection experiments performed suggest a similar incidence of CMFs on TMR and RPR. The CMF cross-section is also similar, but a larger sample of CMFs is necessary to obtain a more precise result. Although the TMR and RPR voters have been hardened in a TMR fashion, voting logic errors can still occur and their incidence increases significantly in higher precision designs, where the voter uses more resources, as is the case of the RPR 64-32-32.

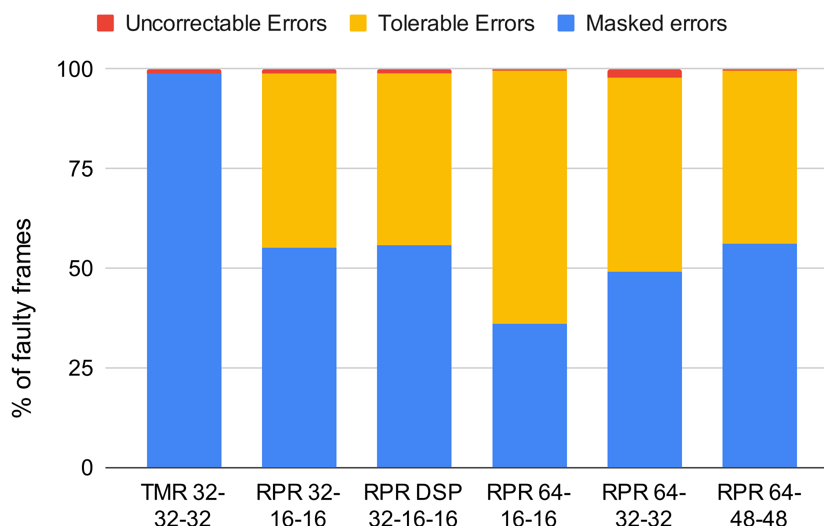


Figure 4.3: Error classification of the faulty frames of the radiation experiments.

Fig. 4.3 is a graphical representation of the error classification of the faulty frames, expressed as a percentage of the total.

It is evident from this figure that the RPR designs have a good correction performance, provided errors smaller than the threshold are tolerated by the system. The impact of the RP size on the designs is also noticeable: the presence of tolerable errors is inversely proportional to the size of the RP modules with respect to the FP module, meaning that reducing the size of the RP modules increases the probability of an error affecting the FP module instead of an RP module, thus forcing the voter to produce a tolerable error. We can clearly see that in those designs were the RP FFT has half the precision of the FP FFT (32-bit FP vs. 16-bit RP or 64-bit FP vs. 32-bit RP), the percentage of tolerable errors is similar to the percentage of masked errors. In other words, the probability of a fault affecting the FP module (producing a tolerable error) is similar to the probability of a fault affecting either of the RP modules (producing a masked error). This effect is even more clear in the RPR 64-16-16 and the RPR 64-48-48: when we decrease the precision of the RP modules, the probability of masked errors decreases harshly, whereas increasing the precision to 48 bits, makes the share of masked errors rise significantly. Again, this is a side-effect of the resources used by the design. Since two 16-bit RP modules use approximately the same amount of LUTs and FFs as one 32-bit FP module (the set of two RP modules uses 55% of the resources dedicated to FFTs, excluding the triplicated

voters), a set of two RP modules is expected to have nearly the same sensitivity as one FP module.

This effect is even more evident with higher precision. For instance, for the case of RPR 64-32-32, where the RP modules represent 52% of the resources needed just for the FFTs, or for RPR 64-16-16, where the RP modules constitute 40% of the resources used for FFT modules in the design.

The estimated cross-section, shown in the next two rows of Table 4.2, has been calculated by dividing the number of erroneous FFT frames by the fluence. The cross-section results prove that the RPR designs are less sensitive to the effects of radiation, hinting that the difference in the cross-section is strongly related to the reduction of the area used by the RPR circuits. Between parentheses we show the confidence intervals of this results with a 95% level of confidence, assuming a Gaussian distribution approximation of the number of faulty frames. Analogously, we calculated a cross-section with respect to the CMFs found and their confidence intervals. These intervals were calculated using the Chi-square distribution, since the number of samples is small. The CMF cross-section shows very similar results in the TMR and RPR designs in their 32-bit versions. The implementations with higher precisions behave differently, although their confidence intervals are wide due to the relatively small size of the samples. The CMF cross-section can be further reduced using several techniques [123]. However, we have not applied any of these techniques in the present work in order to evaluate the raw effects of using RPR irrespective of other considerations.

Regarding the impact of RP size on the sensitivity, it is evident from the cross-section results that making the RP modules more precise, thus making the RPR more alike a TMR, the system is more prone to be affected by radiation effects.

The last row of Table 4.2 contains the average Peak Signal-To-Noise Ratio (PSNR) of the module of the faulty frames. The PSNR is a measure of the distortion produced by errors in a signal. Eq. 4.1 shows how to compute the PSNR for a faulty frame, where MAX is the highest possible value the signal can reach and the MSE is the Mean Squared Error of the FFT frame, calculated as the sum of the squared error of each point in the frame divided by the number of points in the frame.

$$PSNR = 10 \times log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{4.1}$$

For the RPR designs, we computed the PSNR using the voted results that had any kind of errors in them (masked errors yield an infinite PSNR). This is not possible for TMR designs, since the only voted results that contain errors are frames with CMF and those are scarce and have little statistical relevance. For the sake of comparison, we display the average PSNR of the faulty FFT raw results, which should give approximately the same result as an unprotected design. Of course, the majority of the voted results in the TMR design are masked errors and their PSNR would be infinite.

The PSNR results show that the RPR designs yield quite precise results and modifications in the precision of the RP modules also improve the quality of the voted result. In particular, we can observe a significant improvement in the PSNR of the RPR designs when compared to the results of an unprotected design, meaning that even frames with faults yield a better result, thanks to the correction mechanism. We can also see that the designs with increased precision, 64 bit version, have much higher PSNR. This is coherent with the facts that errors affecting the Least Significant Bits of words in these designs should have less importance due to the higher precision, and, for the same reason, the threshold values are smaller, ensuring more precise corrections. The RPR 64-48-48 design, whose RP modules calculate high-precision results, also performs the most precise corrections and has got the highest PSNR.

Table 4.3 presents a simplified error classification of the total faulty data words collected during the irradiation experiments. We define a word as the real or imaginary part of each point in the FFT frame. As we are using 64-point FFTs in these experiments, the result after the voting process of each frame has 128 words. Words are the minimum piece of information correctable by the RPR voters and their analysis is a remarkable source of information. In the first row of Table 4.3 we present the total number of words in the faulty frames found, which is calculated as the number of faulty frames multiplied by 128. In the next row, we show how many of those words are actually affected by errors. In the last rows we classified the erroneous words in three groups. The first group, Masked and tolerable faulty words, gathers faults that did not affect the voted result or affected it slightly, i.e., errors in the FP module smaller than the threshold or significant errors in the FP module that were corrected with the RP results. The second group is made with erroneous voted results in which the faults affected more than one of the FFT results, which are Common-Mode Failures. The last group comprises erroneous voted results that were found even though the three FFT rendered correct results. The latter errors must have happened in the triplicated RPR voters or the majority voter after them and we classified them as Voting logic errors.

From the data in the table, we can see that the erroneous frames of the RPR designs have a smaller share of their words affected by faults (46% in 32-bit TMR versus 28% in the 32-bit RPR). This could be explained by the impact of the less precise data of the RPR, which needs less logic and memory to compute intermediate data and should make the results of these FFTs less sensitive to faults.

When examining the error classification, it is apparent from the data in Table 4.3 that the correction capabilities of the RPR versions of the FFT are on a par with those of the TMR, assuming results with an error smaller than the predefined threshold are tolerated: around 99% of the faults reached the voted output with low levels of noise or no noise at all. As for the masked and tolerable error distribution of the faulty words we observe the same phenomenon that was explained in the faulty frame distribution of Table 4.2. Because the probability of failure of the FFT modules is proportional to the area they use, designs with more precise RP modules with respect to the FP module, thus bigger in size,

produce a higher percentage of masked erroneous words (45% in RPR 64-48-48), whereas smaller RP modules produce a smaller percentage of masked erroneous words (20% in RPR 64-16-16). For RPR 32-16-16 and RPR 64-32-32, the percentage of masked errors was 39% and 33% respectively. The distribution of words with tolerable errors, always caused by a fault in the FP FFT, behaves inversely to the previously explained masked errors.

Table 4.3: CLASSIFICATION OF ERRONEOUS WORDS IN RADIATION EXPERIMENTS

	Unprotected 32-bit FFT	**TMR 32-32-32**	**RPR CLB 32-16-16**	**RPR DSP 32-16-16**	**RPR 64-16-16**	**RPR 64-32-32**	**RPR 64-48-48**
Cross-Section (cm^2)	$0.99\times10^{-}$	$1.82\times^{-10}$	1.49×10^{-10}	1.83×10^{-10}	2.1×10^{-10}	2.64×10^{-10}	2.88×10^{-10}
Total words in faulty frames	27,648	49,408	30,848	25,472	27,392	29,312	38,912
Faulty words (real or imaginary)	12,687 (45.88%)	22,842 (46.23%)	8,625 (27.96%)	6,787 (26.64%)	10,469 (38.22%)	8,849 (30.19%)	12,635 (32.47%)
Masked and tolerable faulty words	-	22,587 (98.88%)	8,483 (98.35%)	6,741 (99.32%)	10,379 (99.14%)	8,766 (99.29%)	12,506 (98.98%)
Common-Mode Failure words	-	252 (1.1%)	119 (1.38%)	33 (0.49%)	90 (0.86%)	7 (0.08%)	0
Voting logic erroneous words	-	3 (0.01%)	23 (0.27%)	13 (0.19%)	0 (0%)	56 (0.63%)	128 (1.02%)

Fault Injection results

In order to obtain more data, we carried out fault injection campaigns on the previously described benchmarks using the Xilinx SEM IP module. The results of these campaigns are compiled in Table 4.4, following the same reasoning used to elaborate Table 4.2. In the first row, we show the number of addresses of the configuration memory that were injected during the experiments. We injected faults in a total of 360,000 randomly selected addresses for every benchmark. The next four rows cover the total amount of faulty FFT calculations obtained and the classification of the frames according to the types of errors found in them. The error classification of the injected results matches very well with the results of the radiation experiments, as evidenced in Fig. 4.4. Only around 1% of the faulty frames contain uncorrectable errors and the share of frames with masked and tolerable errors varies in a similar manner depending on the RPR configuration. The highest percentage of masked errors is found in the design with the lowest ratio of precision between the FP and the RP (RPR 64-48-48). This configuration is the one that is closer to a TMR architecture and thus it is expected that the precision in its output is the highest.

Table 4.4: RESULTS OF INJECTION CAMPAIGNS

	Unprotected 32-bit FFT	TMR 32-32-32	RPR 32-16-16	RPR DSP 32-16-16	RPR 64-16-16	RPR 64-32-32	RPR 64-48-48
Number of injections	360,000	360,000	360,000	360,000	360,000	360,000	360,000
Faulty frames (total)	709	1,907	1,814	1,647	2,053	2,238	2,503
Masked errors	-	1,892 (99.2%)	984 (54.2%)	863 (52.3%)	655 (31.9%)	1,099 (49.1%)	1,440 (57.5%)
Tolerable errors	-	-	817 (45%)	768 (46.6%)	1,382 (67.3%)	1,107 (49.5%)	1,042 (41.6%)
Uncorrectable errors	709	15 (0.8%)	13 (0.8%)	16 (1%)	16 (0.8%)	32 (1.4%)	21 (0.8%)
Error Rate (10^{-3})	1.97 (1.82, 2.10)	5.3 (5.05, 5.53)	5.04 (4.81, 5.27)	4.55 (4.33, 4.77)	5.7 (5.46, 5.95)	6.22 (5.96, 6.47)	6.95 (6.68, 7.23)
CMF Rate (10^{-5})	-	4.2 (2.3, 6.9)	3.6 (1.9, 6.2)	2 (0.8, 4)	4.4 (2.5, 7.2)	8.89 (6.1, 12.5)	5.8 (3.6, 8.9)
PSNR (dB)	68.55	63.15	73.66	73.58	121.26	130.97	153.25

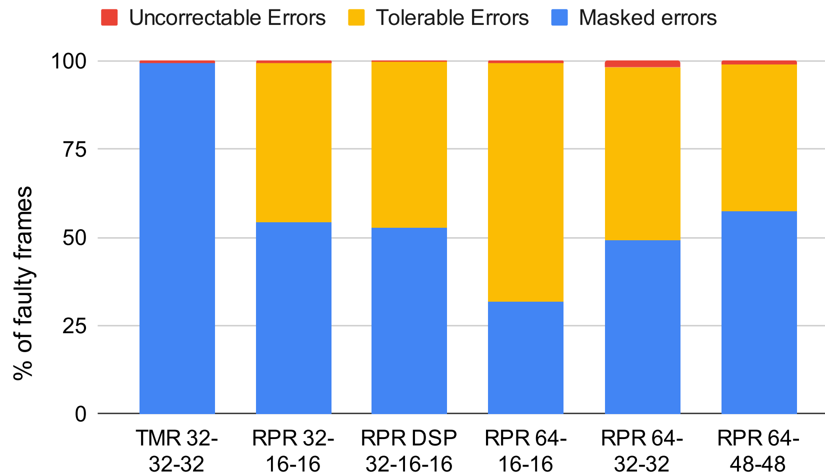


Figure 4.4: Error classification of the faulty frames of the injection experiments.

As for the sensitivity of the circuit to fault injection, we use a figure-of-merit called Error rate, which is calculated as the number of erroneous frames divided by the number of addresses of the configuration memory that were injected. This measure can also be adapted to obtain the CMF rate, which is the rate of CMF frames to the number of injections. Additionally, we computed the confidence intervals of these data using the Gaussian and the Chi-square estimations for the Error and CMF rate, respectively. These rates serve two purposes. First, we can establish a comparison between the injected designs. We can easily see that both the error rate and the CMF rate are lower in those designs that utilize less resources and they seem to be particularly correlated with the usage of LUTs. Second, we can compare the cross-section and the error rate to validate the fault injection results. As represented in Fig. 4.5, a strong correlation exists between the cross-section and the error rate. A similar relationship exists between the CMF cross-section and the CMF rate, but having a small sample space, the trend is not so evident.

Both the error rate and the CMF rate support the idea that increasing the size of the RP modules also increases the sensitivity of the circuit, as it happened in the radiation experiments.

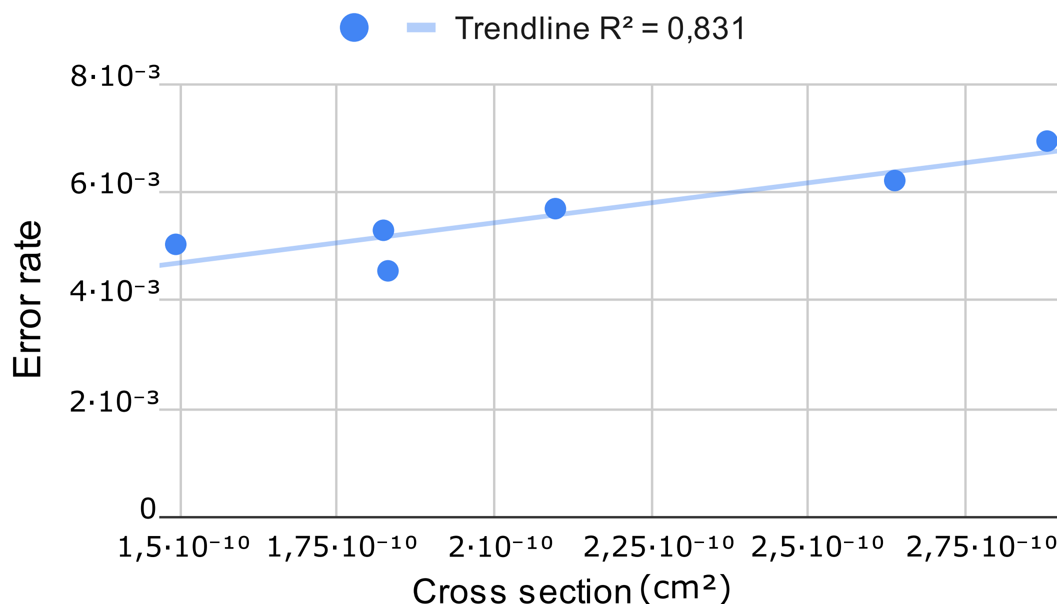


Figure 4.5: Correlation between the Error rate and the Cross-section.

Regarding the PSNR results, shown in the last row of the table, we can observe a similar response of the designs in the radiation and injection experiments, confirming the overall good performance of the error correction mechanism. As it happened in the radiation experiments, the PSNR of the RPR 32-16-16 design evidences a better performance than an unprotected design and the effect of changing the precision of the RP modules is also noticeable in the PSNR. In Table 4.5 we show the same analysis performed in Table 4.3, but for the injection experiments. In this table we show the error classification of the faulty words in the erroneous frames. First, we show the total amount of words evaluated, obtained as the number of faulty frames multiplied by 128, which is the number of words per frame. By comparing each of these words against the expected correct results, we can obtain the number of wrong words in each frame and classify them by their impact on the voted result in the categories previously explained. From the data in the table we can conclude that these results correlate very well with the radiation experiments. The RPR faulty frames present a smaller fraction of their words affected by faults in comparison to TMR, and the percentage of uncorrectable errors matches very well the radiation results, ranging between 1 and 2% of the total words in the faulty frames.

Table 4.5: CLASSIFICATION OF ERRONEOUS WORDS IN INJECTION CAMPAIGNS

	Unprotected 32-bit FFT	TMR 32-32-32	RPR CLB 32-16-16	RPR DSP 32-16-16	RPR 64-16-16	RPR 64-32-32	RPR 64-48-48
Error Rate (10^{-3})	1.97	5.3	5.04	4.55	5.7	6.22	6.95
Total words in faulty frames	90,752	244,096	232,192	210,816	262,784	286,464	320,384
Faulty words (real or imaginary)	37,304 (41.1%)	110,402 (45.23%)	67,284 (28.98%)	60,661 (28.77%)	98,055 (37.31%)	95,028 (33.17%)	108,066 (33.73%)
Masked and tolerable faulty words	-	108,997 (98.73%)	66,040 (98.15%)	59,681 (98.38%)	97,426 (99.36%)	92,997 (97.86%)	106,516 (98.57%)
Common-Mode Failure words	-	1,393 (1.26%)	937 (1.39%)	646 (1.07%)	397 (0.40%)	1,596 (1.68%)	1,372 (1.27%)
Voting logic erroneous words	-	12 (0.01%)	307 (0.46%)	334 (0.55%)	232 (0.24%)	435 (0.46%)	178 (0.16%)

4.6. Conclusions

In this work we carried out radiation experiments and fault injection campaigns to assess the error sensitivity of Reduced Precision TMR approach for several versions of an FFT IP design. The results of this research support the idea that the RPR technique may be used as an alternative to the widespread TMR, with similar correction capabilities under the adequate conditions and similar CMF rate, while reducing the area overhead used by the design.

Our experiments showed that selecting the adequate size for the RP instances of the design is key to achieve a satisfactory result. A high-precision RPR produces less noise in the output of the circuit when a correction is enforced, but also makes the circuit more sensitive to errors, as it requires more hardware resources. On the contrary, a low-precision RPR would make the design less sensitive, but the correction would be less precise.

We used the same setup for fault injection and radiation testing with low energy protons, and we have shown that the results are highly correlated. Thus, fault injection can be used to obtain additional data for rare events or to perform a preliminary evaluation of an RP-TMR design before testing it under the beam.

References

[160] M. Choudhury and K. Mohanram, "Low cost concurrent error masking using approximate logic circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 8, pp. 1163-1176, Aug. 2013.

[161] J. Sanchez-Clemente, L. Entrena, R. Hrbacek, L. Sekanina. "Error Mitigation using Approximate Logic Circuits: A Comparison of Probabilistic and Evolutionary Approaches". IEEE Transactions on Reliability, vol. 65, no. 4, pp. 1871-1883, Sep. 2016.

[162] A. J. Sánchez Clemente, L. Entrena y F. Kastensmidt, "Approximate TMR for

selective error mitigation in FPGAs based on testability analysis," 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 112-119, Aug. 2018.

[97] B. Shim and N. R. Shanbhag, "Reduced Precision Redundancy for Low-power Digital Filtering," Proc. 35th Asilomar Conference on Signals, Systems and Computers, pp. 148-152, vol. 1, 2001.

[98] B. Shim and N. R. Shanbhag, "Energy-Efficient Soft Error-Tolerant Digital Signal Processing," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 4, pp. 336-347, Apr. 2006.

[168] B. Pratt, M. Fuller and M. Wirthlin, "Reduced-Precision Redundancy on FPGAs," Int. Journal of Reconfigurable Computing, vol. 2011, Article ID 897189, 2011.

[170] S. Liu, K. Chen, P. Reviriego, W. Liu, A. Louri and F. Lombardi, "Reduced Precision Redundancy for Reliable Processing of Data," IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 4, pp. 1960-1971, Oct.-Dec. 2021.

[122] L. Sterpone and M. Violante, “A new reliability-oriented place and route algorithm for SRAM-based FPGAs,” IEEE Transactions on Computers, vol. 55, no. 6, pp. 732–744, Jun. 2006.

[146] H. Quinn, K. Morgan, P. Graham, J. Krone, M. Caffrey, and K. Lundgreen, “Domain crossing errors: Limitations on single device triple-modular redundancy circuits in Xilinx FPGAs,” IEEE Transactions on Nuclear Science, vol. 54, no. 6, pp. 2037–2043, Dec. 2007.

[123] M. J. Cannon, A. M. Keller, H. C. Rowberry, C. A. Thurlow, A. Pérez Celis and M. J. Wirthlin, "Strategies for Removing Common Mode Failures From TMR Designs Deployed on SRAM FPGAs," IEEE Transactions on Nuclear Science, vol. 66, no. 1, pp. 207-215, Jan. 2019.

[163] A. J. Sánchez-Clemente, L. Entrena and M. García-Valderas, "Partial TMR in FPGAs Using Approximate Logic Circuits," IEEE Transactions on Nuclear Science, vol. 63, no. 4, pp. 2233-2240, Aug. 2016.

[94] M. Berg, “Single Event Effects in FPGA Devices 2014-2015”, NASA Electronic Parts and Packaging Program (NEPP) Electronics Technology Workshop (ETW), June 2015.

[8] L. A. García-Astudillo, A. Lindoso, M. Portela and L. Entrena, "Evaluation of a Reduced Precision Redundancy FFT Design", 2020 XXXV Conference on Design of Circuits and Integrated Systems (DCIS), pp. 151-156, Nov. 2020

[164] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," 18th IEEE European Test Symposium (ETS), pp. 123-128, May 2013.

[166] A. M. Keller and M. J. Wirthlin, "Partial TMR for Improving the Soft Error Reliability of SRAM-Based FPGA Designs", IEEE Transactions on Nuclear Science, vol.

68, no. 5, pp. 1023-1031, May 2021.

[138] Xilinx Inc., “Zynq-7000 SoC Datasheet: Overview”, Datasheet DS190 (v1.11.1), Jul. 2018.

[74] Xilinx Inc., “Soft error mitigation controller v4.1,” Product guide PG036, Apr. 2018.

[171] Xilinx Inc., “Fast Fourier Transform v9.1,” Product Guide PG109, Jun. 2020.

[172] A. Lindoso, M. García-Valderas, L. Entrena, Y. Morilla, P. Martin-Holgado. "Evaluation of the suitability of NEON SIMD microprocessor extensions under proton irradiation". IEEE Transactions on Nuclear Science, vol. 65, no. 8, pp. 1835-1842, Apr. 2018.

[165] A. Bosio, I. O’Connor, G. S. Rodrigues, F. K. Lima and S. Hamdioui, "Exploiting Approximate Computing for implementing Low Cost Fault Tolerance Mechanisms," 15th Design & Technology of Integrated Systems in Nanoscale Era (DTIS), pp. 62-63., Apr. 2020.

[167] G. S. Rodrigues, J. S. Fonseca, F. L. Kastensmidt, V. Pouget, A. Bosio, and S. Hamdioui, “Approximate TMR based on successive approximation and loop perforation in microprocessors”, Microelectronics Reliability, vol. 100–101, 113385, Sept. 2019.

[169] G. S. Rodrigues, J. Fonseca, F. Benevenuti, F. Kastensmidt and A. Bosio, "Exploiting Approximate Computing for Low-Cost Fault Tolerant Architectures," 32nd Symposium on Integrated Circuits and Systems Design (SBCCI), article no. 3, pp. 13-18, Aug. 2019.

5. ANALYZING SCALED REDUCED PRECISION REDUNDANCY FOR ERROR MITIGATION UNDER PROTON IRRADIATION

Using the Fast Fourier Transform benchmarks we developed for the experiments presented in Chapter 3, we proposed a modified version of the Reduced Precision Redundancy technique as a way to improve the loss of precision in the corrected results. These modified versions of the FFT benchmarks perform precision reduction only on the first stages of the pipeline, which, coincidentally, are the ones that use the most resources. This way, the precision of the corrected results is improved at a small increase of resources.

For comparison purposes, we used the previously tested BTMR designs as well as newly developed standard Reduced Precision Redundancy benchmarks for the different FFT benchmarks. For all of them we performed exhaustive fault injection campaigns and proton irradiation experiments for the ones that we considered more relevant.

This chapter has been published as an article:

[3] © 2022 IEEE. Reprinted, with permission, from L. A. Garcia-Astudillo, A. Lindoso, L. Entrena, *et al.*, "Analyzing Scaled Reduced Precision Redundancy for Error Mitigation under Proton Irradiation," *IEEE Transactions on Nuclear Science*, pp. 1–1, 2022. DOI: 10.1109/TNS.2022.3147599

Abstract

Reduced Precision Redundancy (RPR) is an alternative to Triple Modular Redundancy (TMR) that reduces the area overhead at the expense of minor accuracy loss in case of error. In this work, we propose a Scaled RPR approach for multi-stage circuits and analyze the error mitigation tradeoffs. As case study, several Fast Fourier Transform designs were tested with low energy protons and fault injection. Experimental results show that the proposed approach achieves error mitigation with good accuracy, while significantly reducing the area overhead with respect to a full precision TMR approach.

5.1. Introduction

Triple Modular Redundancy (TMR) is a widely used approach to mitigate errors produced by radiation in digital circuits. TMR consists in triplicating the logic and adding majority voters to mask an error that may appear in one of the three replicas. Triplication may be applied selectively to some critical elements, such as flip-flops, to mitigate the effects

of Single-Event Upsets (SEU), or it may be eventually extended to the entire circuit for higher protection. Full triplication can be very effective to mitigate Single-Event Effects (SEEs), as long as the replicas are conveniently isolated to avoid Common Mode Failures (CMFs), i.e., errors affecting more than one replica at the same time [123]. However, it also results in a very high overhead.

In the case of Application Specific Integrated Circuits (ASICs), TMR is usually applied only to flip-flops, while other SEU sensitive elements, such as memories, can be protected by Error Detection And Correction (EDAC) codes. Single-Event Transients (SETs) can be mitigated by TMR, but this requires triplicating the combinational logic. Thus, alternative solutions with lower overheads, such as SET filters [173], are often preferred. In SRAM-based Field Programmable Gate Arrays (FPGAs), triplication must be applied to the entire circuit, including the combinational logic and the routing, because the behavior is defined by a SEU sensitive configuration memory [94]. Careful design must be performed to avoid errors in the voting logic, so that the voters themselves are generally triplicated as well. Even though TMR has been proved to be cost effective for SRAM-based FPGAs, the overhead it produces is very high and can require up to six times the area of the circuit [145]. This overhead carries a significant negative impact on power consumption and radiation sensitivity as well.

Attempts to reduce the overhead are based on temporal redundancy [174] and partial or approximate redundancy [175]-[168]. Approximate computing has emerged as a way to meet the performance and power consumption challenges of modern computing systems. In many applications, particularly those related with large data processing, such as image processing, communications, digital signal processing, etc., achieving high accuracy is a relative goal that must be balanced with other requirements. In this case, approximate computing techniques seek to achieve high energy efficiency and performance within acceptable accuracy loss. Approximate TMR (ATMR) techniques have already been proposed that use approximate versions of the circuit instead of exact replicas to save resources [175], [163]. Reduced Precision Redundancy (RPR) is a general approximation approach that consists in reducing the precision of the operations in the redundant replicas [97], [98], [170]. RPR has been shown to provide a good tradeoff for digital signal processing applications implemented in FPGA [168].

RPR-based TMR is built with the original circuit and two redundant modules with reduced precision. A special voter must be used, as the three circuits are not exactly the same. The redundant modules typically use a common and constant reduced precision for simplicity. However, it is possible to use different precisions for each processing stage [105].

In this work we analyze the effectiveness of using a Scaled RPR in the TMR design. We use the Fast Fourier Transform (FFT) as a case study. The FFT is a well-known algorithm that is widely used in digital signal processing and is a representative benchmark that uses logic, arithmetic, and storage resources. The algorithm is divided

into several stages that can be executed in pipeline mode for real-time processing. Rather than using a common reduced precision for the redundant modules, we scaled the precisions for the different stages so as to ensure that SEU effects are either masked or result in an acceptable loss of accuracy. Thus, a better balance can be obtained by using more aggressive precision reduction in the early stages which have less impact on the total accuracy of the system. This approach offers similar mitigation capabilities than a full scaled approach, but much more accurate results, while still reducing the overhead significantly with respect to a full precision TMR approach. To analyze the Scaled RPR approach, an irradiation campaign has been carried out using low energy protons on a 28 nm technology FPGA. Additionally, a fault injection campaign has also been performed as a complementary means to validate the proposed approach.

The rest of this paper is organized as follows. Section 5.2 describes the scaled architectures used in this work. Section 5.3 describes the experimental setup. Section 5.4 discusses the results of the experiments. Finally, Section 5.5 summarizes the conclusions of this work.

5.2. Scaled FFT architectures

The usual approach to Reduced Precision Redundancy (RPR) uses two Reduced Precision (RP) modules to protect a Full Precision (FP) module [97]. These RP modules are implemented as smaller copies of the FP module, whose least significant bits are truncated or rounded off. Using a specially tailored voting logic, an error in the FP module can be partially corrected with the approximate result from the RP modules. More precisely, the output of the FP module is taken unless it differs significantly from the output of the two RP modules, provided that the latter produce exactly the same result. This voting approach ensures that large errors in the FP module are masked and that the output is either correct or at most contains tolerable inaccuracies. This way, a system that can tolerate a certain level of noise in its output can be hardened against faults with a lower area overhead than a TMR implementation. Fig. 5.1 illustrates this technique applied to harden a Fast Fourier Transform module [1].

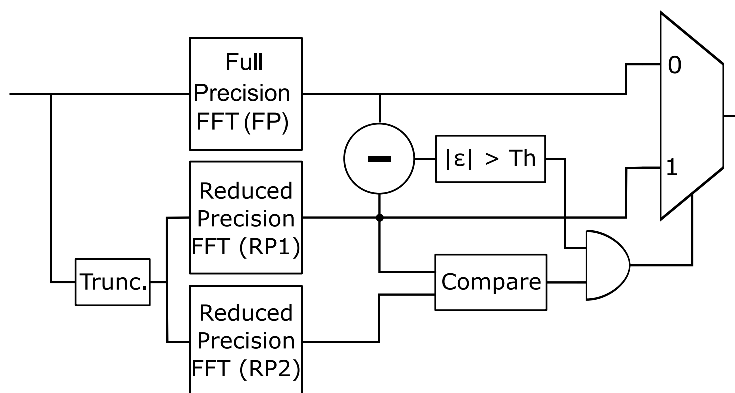


Figure 5.1: Block diagram of a Reduced Precision Redundancy hardening method.

This approach has been successfully tested on serial FFT implementations [8] [1]. Nevertheless, the traditional RPR would be inefficient in a pipeline implementation for two reasons. First, reducing the bits used by all stages of the pipeline would result in a damped and less precise output. Second, the last stages of the pipeline use less resources than the first ones, so reducing the precision of those stages only adds a marginal benefit. Liu et al. proved that scaling the word length in the first stages of a pipeline FFT architecture significantly helps in lowering the power and resource consumption of the design, while producing a low-noise output [105].

One of the goals of this study is to demonstrate that the traditional RPR approach, that worked very well in serial FFT implementations, is not as efficient for pipeline architectures. For this reason, we followed the same methodology and metrics used in [1], but applied to pipeline FFT designs. The scaled FFT approach is proposed as an alternative to traditional RPR for pipeline architectures because of its higher precision in the output with intermediate consumption of resources. The main goal of this study is assessing the performance of the scaled RPR hardening technique. For this reason, we also implemented TMR and traditional RPR versions of the pipeline FFT design for comparison.

In this work, we consider Radix-2 (R2SDF), Radix-4 (R4SDF) and Radix-2^2 (R2^2SDF) FFT pipelined architectures whose stages can be individually scaled to operate with the desired word length. For our case study, we configure them to compute 256-points FFT frames and each complex point has 16-bit real and imaginary parts by default. These designs follow the Single-Path Delay Feedback (SDF) approach [157], which uses FIFO memories to delay intermediate data and feed them to the Butterfly Units (BU) when necessary.

Fig. 5.2 describes the architecture of a Radix-2 stage, which is comprised by a two-input BU, a FIFO memory, and a complex multiplier. Each stage has a single input data stream and a single output data stream, but the BU operates on two inputs and produces two outputs in each clock cycle. The FIFO memory is used to delay part of the input and output data streams in order to produce a double input data stream for the BU and a single output data stream for the multiplier. The FIFO memory for the first stage is capable of storing N/2 data, where N is the transform length. The size of the FIFO is halved in each subsequent stage.

This FFT architecture uses log2(N) of these stages. Fig. 5.2 illustrates a Radix-4 stage, consisting of a four-input BU, three FIFO memories and a multiplier unit. The FIFOs have the same depth within the stage, but they are larger in the first stages. Fig. 5.2 details the structure of a Radix-2^2 stage, which includes two different BUs, one FIFO per BU and one complex multiplier. In the first stage, the two FIFOs hold N/2 and N/4 data points, and these values are halved in each subsequent stage. The Radix-4 and Radix-2^2 architectures use log4(N) stages.

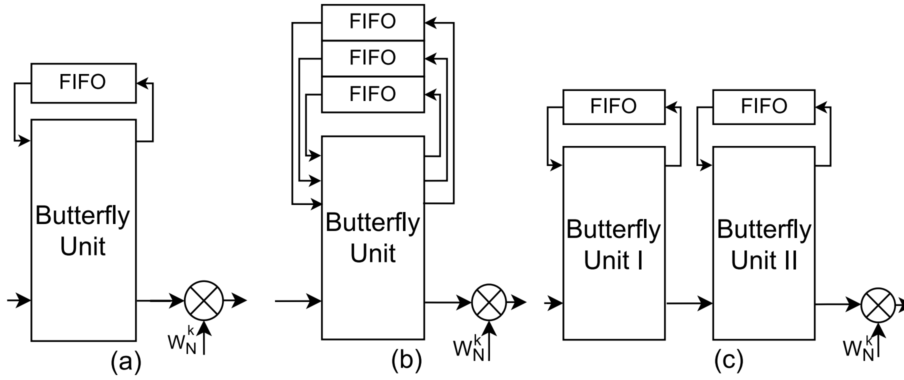


Figure 5.2: Stages in FFT architectures: (a) Radix-2, (b) Radix-4, (c) Radix-2^2.

In order to select an adequate scaling pattern for our designs, we developed an RPR system and set a threshold value of 2-8 (0.39%) for a default 16-bit word length. In contrast, an RPR whose stages are equally scaled would typically require at least a 2-6 (1.6%) threshold to avoid false positives. Then, we proceeded to adjust the word length of the different stages to achieve an output whose difference with the full precision reference would not exceed the threshold value in simulations. The scaling of the stages was iteratively selected using an Area Limited Design (ALD) algorithm [105]. This means reducing the size of the first stage until the threshold is surpassed, then increasing the size by one and repeating the process with the next stage until all of them are at their minimum value allowed by the threshold. With this procedure, we selected a [8, 9, 10, 11, 12, 16, 16, 16] scaling pattern for the R2SDF design and a [8, 10, 12, 16] pattern for both the R4SDF and R2^2SDF architectures, as shown in Fig. 5.3 and 5.3 respectively. It can be easily seen that the ALD algorithm converges quite fast and only about half the stages need scaling. This, along with the fact that the number of steps in the pipeline follows a logarithmic scale, makes the ALD algorithm a feasible approach even with longer FFT frames.

By scaling in this way, we achieved a significant resource reduction. Fig. 5.4 compares the usage of various resources between a full precision and a scaled R2^2SDF FFT.

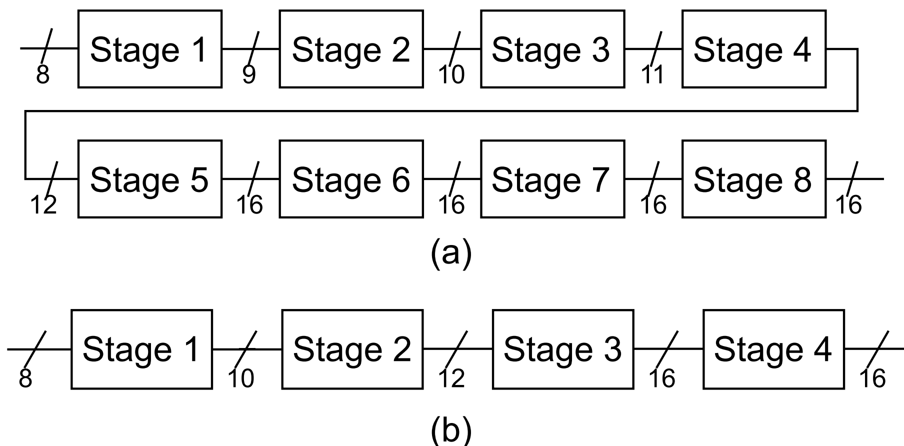


Figure 5.3: Scaling pattern in FFT pipelined architectures: (a) Radix-2 (b) Radix-4 and Radix-2^2.

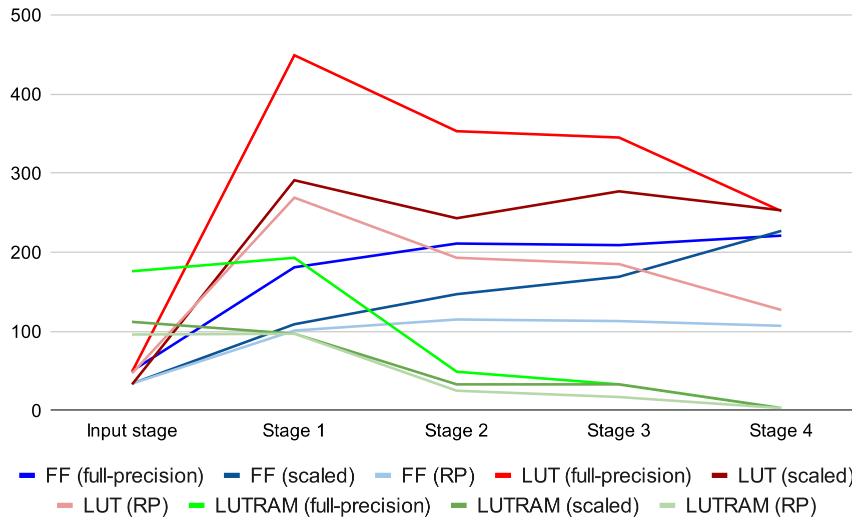


Figure 5.4: Resource utilization of the stages of full precision, reduced precision and scaled reduced precision Radix-2^2 FFTs.

5.3. Experimental setup

Based upon the previously described scaled FFTs, we implemented a set of circuits using the different FFT architectures hardened with TMR, traditional RPR and Scaled RPR. The set comprises TMR, RPR and Scaled RPR versions of 256-points Radix-2, Radix-4, and Radix-2^2 pipelined architectures. The Full Precision (FP) module uses a word length of 16 bits for real and imaginary data. The traditional RPR version uses RP modules with a word length of 8 bits in every stage. These two sets of architectures are tested to put into perspective the performance of the scaled RPR. The Reduced Precision (RP) FFTs in the RPR designs are scaled versions of the architecture, using the scaling patterns described in the previous section. This way, the scaled RPR versions achieve a significant area reduction with low noise in the outputs of the RP modules.

The synthesis results shown in Fig. 5.5 correspond to the resources used by the different designs tested in this work. These results include the area used by the three FFT blocks, as well as the FFT control logic and the triplicated voting circuits. A notable reduction of resources in the scaled RPR is apparent from the graph: up to 10% in the number of flip-flops, 9% in the number of LUTs implemented as logic and 14% in LUT as RAM. The actual reduction in the scaled FFT modules is around 20% for all resources when compared to the unscaled version, but the complexity of the RPR voter reduces the positive impact of the scaling on the final area overhead of the system. Nevertheless, the reduction of resources achieved by the scaled RPR is not as significative as in the RPR designs, which achieve reductions up to 33% in some architectures at the cost of less precision in the RP modules.

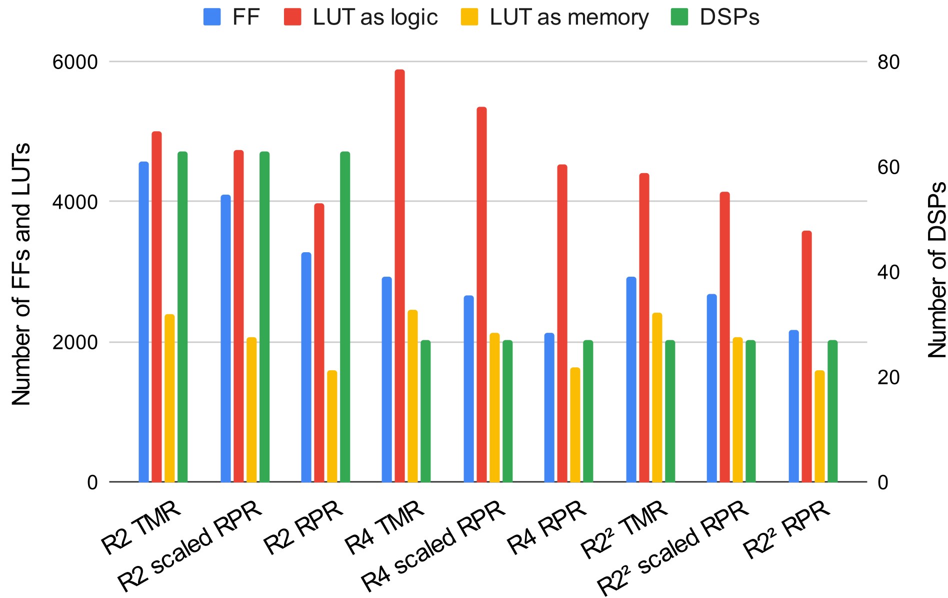


Figure 5.5: Resource utilization of the different designs.

The objective of the conducted radiation experiments is twofold: firstly, determine the error mitigation of RPR using scaling in the Reduced Precision modules compared to a full precision TMR implementation and a traditional RPR; and secondly, establish a comparison between FFT architectures to allow a deeper insight into which architecture performs better under radiation.

To control the experiments, each design was wrapped in a testbench circuit able to provide input data for the FFTs, compare the outputs against the expected results to detect errors and collect data for later analysis. The results from the three FFTs and the voted output are collected upon error and sent through a UART connection to an external device that hosts the experiments.

A Xilinx Zynq-7010 All Programmable SoC (APSoC) has been used to implement the designs for these experiments. Although this device includes a dual core ARM processor, only the programmable logic was used in the implementation.

We carried out one irradiation campaign at Centro Nacional de Aceleradores (CNA) in Seville with low-energy protons. The experiments used an average energy of 15.2 MeV and an average flux of 4.3×108 p+/cm2s. With the same objectives of the radiation experiments, we also performed a fault injection campaign using Xilinx Soft Error Mitigation (SEM) IP [74]. The SEM IP is capable of detecting and correcting errors in the configuration memory of the FPGA and can also be configured to inject faults. Faults were injected at random addresses and with a rate that is slow enough to guarantee a complete FFT calculation between injections. The same configuration can be employed in irradiation experiments by disabling the SEM IP. The injection results can also be compared with the radiation data to confirm its validity.

We used default placement strategies for all the designs, not aiming to study their incidence on Multiple Cell Upsets (MCU) or Common-Mode Failures (CMF), and all the FFTs shared the same clock domain.

5.4. Experimental results

Radiation results

Table 5.1 shows the results of the irradiation campaign. We report two versions of the scaled RPR architectures, namely R2SDF and $R2^2$SDF FFT architectures, as well as three other $R2^2$SDF designs for comparison purposes: a full precision TMR, a full precision unprotected design and a scaled unprotected design. The first two rows show the total amount of erroneous FFT calculations (Faulty frames) found and the number of those classified as Uncorrectable frames, respectively. We define an Uncorrectable frame as a frame in which a fault affected simultaneously more than one FFT module (Common-Mode Failure), the voting logic or a combination of those in at least one of the points in the frame. All these events produce errors that cannot be corrected by the voting logic.

Based on the number of Faulty frames and Uncorrectable frames, and data from the proton beam, we calculated the cross-sections found in the next two rows. The cross-section is calculated as the number of events found divided by the fluence of the proton beam. Additionally, between parentheses we show the confidence intervals of these calculations. For the confidence intervals of the cross-section, we used a Gaussian distribution approximation with a level of confidence of 95%. Since the number of Uncorrectable events is small, the confidence intervals of the Uncorrectable cross-sections were calculated assuming a Chi-square distribution of the Poisson mean with the same confidence.

The next rows present an error classification based on the faulty words found. We define a word as the minimum correctable portion of information, which in this case refers to the real or imaginary part of each of the complex points in an FFT frame. There are 512 words in each 256-points FFT frame.

Faulty words can be classified in two groups. First, Correctable errors, which include mitigated errors and, in the case of RPR, also include errors below the tolerable threshold as well as errors partially corrected with the approximate result. The second group is Uncorrectable words, that reports faults affecting more than one FFT at the same time, the voting logic, or a combination of these cases. Fig. 5.6 reports the error classification of the faulty words found in the faulty frames of the radiation experiments. By analyzing this graph, we can conclude that the incidence of uncorrectable errors is similar for the three designs. We can also see that the percentage of masked errors is slightly higher in the Radix-2 architecture than in the Radix-2^2 architecture. This result can be explained by

the higher usage of resources of the Radix-2 scaled RP FFTs, making them more sensitive and, thus, leading to more masked errors.

In the last row of the table, we present the Peak Signal-to-Noise Ratio (PSNR), that is a measure of the distortion in a signal caused by the radiation errors. PSNR is calculated as the ratio between the maximum energy peak of the signal and the Mean-Squared Error (MSE), expressed as in Eq. 5.1. For the PSNR calculations we used the voted results of the faulty frames in the RPR architectures. In the TMR architectures, however, we could not use the faulty voted results, because those correspond to CMF errors and that PSNR would not be statistically significant (the average PSNR of TMR voted results is infinite). Instead, we use the results of the faulty frames prior to the voting. This value cannot be compared directly to the RPR PSNRs but offers a good measure of the PSNR of unprotected FFTs.

$$PSNR = 10 \times log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{5.1}$$

Table 5.1: RESULTS FROM THE IRRADIATION EXPERIMENTS

	R2SDF	R2^{2}SDF			
	Scaled RPR	TMR	Scaled RPR	Unprotected	Scaled unprotected
Faulty frames	187	180	176	91	99
Uncorrectable frames	4	4	4	-	-
Cross-section (10^{-10} cm^2)	3.0 (2.6, 3.5)	1.9 (1.6, 2.1)	2.8 (2.3, 3.2)	1.1 (0.8, 1.3)	1.0 (0.8, 1.3)
Uncorrectable cross-section (10^{-12} cm^2)	9.6 (3.6, 21.1)	4.1 (1.1, 10.6)	10.9 (4.4, 22.5)	-	-
Total words in faulty frames	95744	92160	90112	46592	50688
Faulty words	17145 (17.9%)	28166 (30.6%)	17758 (19.7%)	11336 (24.3%)	28867 (59.3%)
Correctable (%)	94	93	93.8	-	-
Uncorrectable words (%)	6	7	6.2	-	-
PSNR (dB)	90.6	81.7	90.5	80.6	73.4

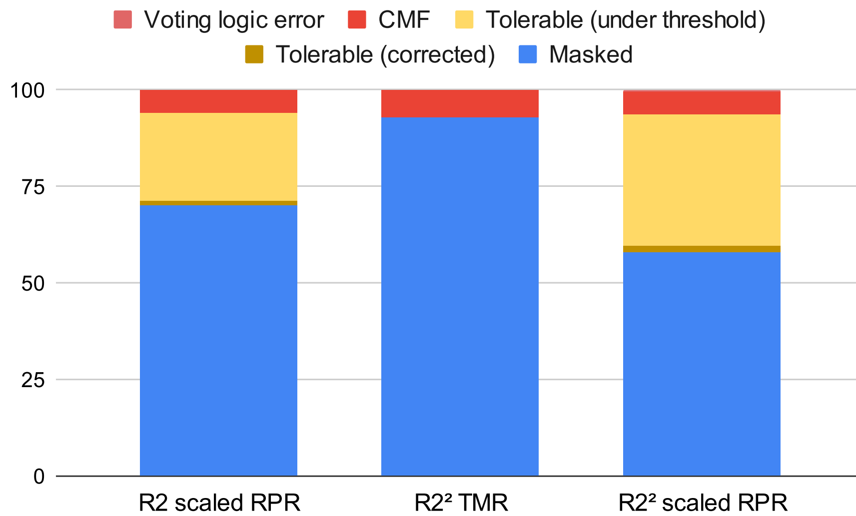


Figure 5.6: Error classification of the faulty words found in the radiation experiments.

According to the data in Table 5.1, the three hardened designs have a similar cross-section, while the unprotected designs are around three times less sensitive than the hardened designs because they use fewer resources. The hardened designs have been equally affected by CMFs and voting logic errors in this experiment. What stands out in the performance of the RPR systems is the reduction in the fraction of words affected by faults in a frame, which also influences the increase in the average PSNR of the RPR designs. While having nearly the same percentage of correctable and uncorrectable words than the TMR design, having less faulty words results in a more precise result. A possible explanation for this behavior could lay on the scaling process: errors affecting a word in the output of one of the first stages may be removed by the truncation in the next stage, attenuating the impact of the error or even masking it completely if the bit flips occurred in the LSBs of the word. This mechanism may be responsible for the diminishing of the percentage of faulty words per frame as well as the enhancement of the quality in the final output of the FFT.

Injection results

We also performed fault injection campaigns that complement the data obtained from the irradiation tests. These tests are aimed at obtaining a more thorough analysis of the designs and assessing additional designs. The fault injection results are summarized in Table 5.2 and Table 5.3. By analyzing these results, we want to establish a correlation between the radiation and the injection campaign. Additionally, thanks to longer experiments, we want to detect significant differences in the number of faulty frames and frames affected by uncorrectable errors among the different FFT architectures and hardening techniques.

The first four rows of these tables present the number of faulty FFT calculations found during the experiments and, of those, the number of frames affected by uncorrectable errors, this is, CMF errors and voting logic errors. To normalize the data across the table with the number of addresses injected, we calculated the Error rate and the Uncorrectable rate as the number of faults found per 1000 injections. Analogously to the irradiation campaign, we calculated the confidence intervals of the Error rate and Uncorrectable rate using the Gaussian and Chi-square approximations, respectively, with a 95% level of confidence.

The rest of the rows in the table classify the erroneous words in the same way as Table 5.1 and we report the PSNR in the last row.

Table 5.2: RESULTS FROM THE FAULT INJECTION EXPERIMENTS IN RADIX-2 AND RADIX-4 ARCHITECTURES

	R2SDF			R4SDF		
	TMR	RPR	Scaled RPR	TMR	RPR	Scaled RPR
Number of injections	360000	360000	360000	360000	360000	360000
Faulty frames	1662	1423	1565	1740	1544	1644
Error rate (×10^{-3})	4.62 (4.4, 4.8)	3.95 (3.7, 4.2)	4.35 (4.1, 4.6)	4.83 (4.7, 5.2)	4.29 (4.1, 4.5)	4.57 (4.3, 4.8)
Uncorrectable frames	51	66	47	32	38	36
Uncorrectable rate (×10^{-3})	0.14 (0.11, 0.19)	0.18 (0.14, 0.23)	0.13 (0.10, 0.17)	0.09 (0.06, 0.13)	0.11 (0.08, 0.14)	0.10 (0.07, 0.14)
Words in faulty frames	850944	728576	801280	890880	790528	841728
Faulty words	269807 (31.7%)	133201 (18.3%)	159171 (19.9%)	216161 (24.3%)	130327 (16.5%)	146338 (17.4%)
Correctable (%)	93.6	93.5	91.3	95.8	94.6	91
Uncorrectable (%)	6.4	6.5	8.7	4.2	5.4	9
PSNR (dB)	77.3	79.9	86.9	82.1	85.9	91.2

Table 5.3: RESULTS FROM THE FAULT INJECTION EXPERIMENTS IN RADIX-2^2 ARCHITECTURE

	R2^{2}SDF				
	TMR	RPR	Scaled RPR	Unprotected	Scaled unprotected
Number of injections	360000	360000	360000	360000	360000
Faulty frames	1585	1441	1531	423	516
Error rate (×10^{-3})	4.4 (4.2, 4.6)	4.00 (3.8, 4.2)	4.25 (4.0, 4.5)	1.17 (1.1, 1.3)	1.43 (1.3, 1.6)
Uncorrectable frames	41	67	47	-	-
Uncorrectable rate (×10^{-3})	0.11 (0.10, 0.18)	0.19 (0.14, 0.24)	0.13 (0.10, 0.17)	-	-
Words in faulty frames	811520	737792	783872	216192	264064
Faulty words	260484 (32.1%)	138921 (18.8%)	148157 (18.9%)	194968 (22.5%)	303779 (28.8%)
Correctable (%)	94.6	89.4	87.3	-	-
Uncorrectable (%)	5.4	10.6	12.7	-	-
PSNR (dB)	78.8	82.5	89.4	81.1	67.2

From these data, we can confirm a lower error rate in all the scaled RPR designs with respect to the ones hardened by TMR using the same FFT architecture. As expected, the error rate of the traditional RPR is the lowest of the three for all architectures because of its lower consumption of resources. Among the different architectures, those that use less resources are less sensitive to failure. Fig. 5.7 shows that a clear trend exists with this respect. We can clearly establish that the scaled RPR technique is effective in reducing the error rate in all the architectures when compared with the TMR. As expected, its performance is not as good as the traditional RPR in these terms because of the higher area usage.

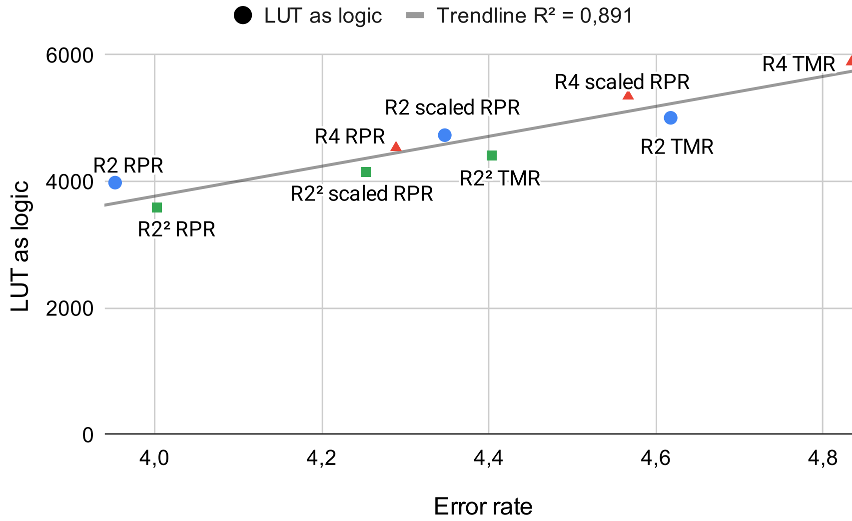


Figure 5.7: Correlation between the Error rate and the usage of LUTs as logic.

However, the incidence of Uncorrectable errors is similar for all cases, with a slight decrease in the Radix-4 architecture. The percentage of faulty words in the RPR designs and the PSNR matches very well the data from the irradiation campaign, which reinforces the idea of the correction of errors due to truncation. The percentage of affected words in the RPR designs, both scaled and traditional, is smaller than in the TMR designs.

Although the number of samples is small, our data suggest a good correlation between the cross-section calculated in the irradiation campaign and the Error rate from the injection campaign, as seen in Fig. 5.8.

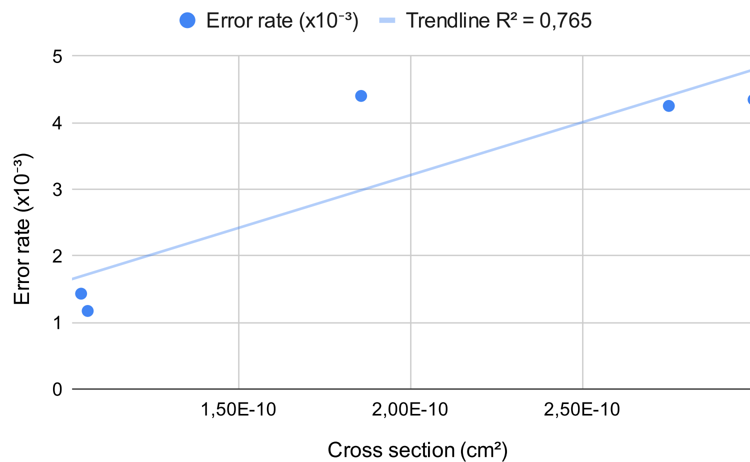


Figure 5.8: Correlation between the Cross section and the Error rate.

Attending to the error classification of these erroneous words in the faulty frames, which we can see in Fig. 5.9, the percentage of CMF words is generally lower in TMR designs, although the total amount of errors is similar within the same architecture for the three hardening methods. The percentage of masked errors in the scaled RPR designs is

greater than in the traditional RPR and that seems to be a logical result. Because the RP modules in the scaled RPR use more resources than the RP modules in the RPR designs, those RP FFTs should be more sensitive to failure, producing masked errors.

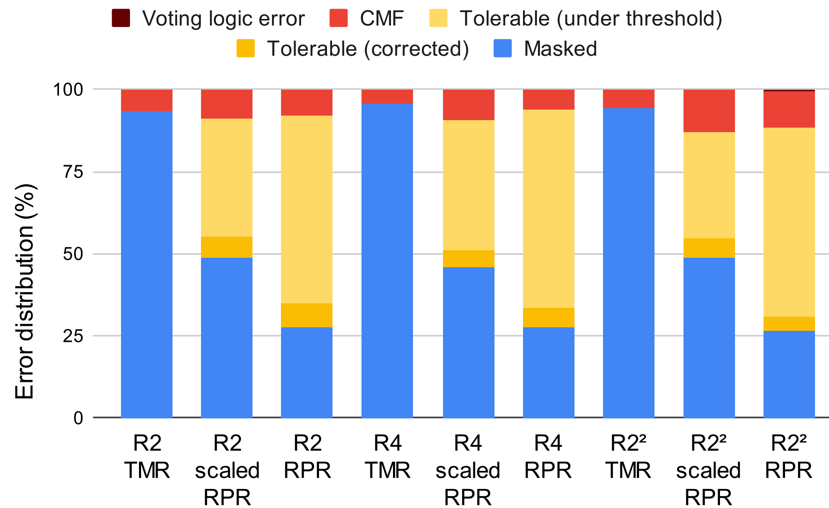


Figure 5.9: Error classification of the faulty words found in the frames.

The PSNR results of the injection test support the results of the irradiation campaign and, as expected, the precision of the scaled RPR designs is better than the precision of the traditional RPR, which makes sense, because of the scaled approach in the precision of the stages of the FFT.

Comparison with serial FFT implementations

A comparison of the results from this study with previous results presented in [1] using serial implementations of the Radix-4 FFT provides some interesting outcomes.

Overall, we can observe that the Radix-4 pipeline implementations of the present study have a slightly lower error rate. However, the incidence of Uncorrectable errors is almost twice as important for all the pipeline architectures. In both this study and [1], we found a strong correlation between the usage of LUTs and the error rate within the same architecture. This, however, does not seem to apply when comparing serial and pipeline architectures, since the pipeline architectures in this work use almost twice more LUTs than their serial equivalents, but have a lower error rate.

This contradiction could be attributed to the way calculations are performed in the pipeline architectures and their intrinsic correction capabilities. In a pipeline architecture, routing or configuration errors in one stage of the pipeline would only affect the results of the next stages and the scaling could even mitigate them if they only affect the least significant bits. On the contrary, in the serial implementations, due to the reutilization of resources, every iteration of the FFT would be flawed, producing a higher number

of faulty calculations as well as lower quality results. Regarding the quality, we can clearly see that the pipeline implementations boast of a higher PSNR, around 10 dBs higher for every design, than the serial implementations. In particular, the scaled RPR implementations have the highest PSNR.

The rate of Uncorrectable errors may not be much affected by how the calculations are made. On the contrary, the amount of resources and how they are laid out in the FPGA can have a big influence. Having used default placement strategies for both the serial and pipeline architectures, we can see that the increase in the number of uncorrectable FFT calculations is correlated with the increase in the number of LUTs needed by the pipeline architectures.

5.5. Conclusions

In this work, we proposed a novel RPR approach for pipeline architectures that uses Reduced Precision modules whose stages are scaled individually. We used proton irradiation and fault injection campaigns to validate the proposed hardening technique with different FFT implementations and compared its performance with other popular mitigation techniques (TMR and conventional RPR). This approach achieves error mitigation with good precision, while significantly reducing the area overhead with respect to a full precision TMR approach. Reducing the use of resources also lowers the sensitivity of the circuit to radiation, and due to the scaling, the number of erroneous points per FFT calculation is significantly reduced, minimizing the noise in the voted output.

References

[123] M. J. Cannon, A. M. Keller, H. C. Rowberry, C. A. Thurlow, A. Pérez Celis and M. J. Wirthlin, "Strategies for Removing Common Mode Failures From TMR Designs Deployed on SRAM FPGAs," IEEE Trans.on Nuclear Science, vol. 66, no. 1, pp. 207-215, Jan. 2019.

[173] D. G. Mavis and P. H. Eaton, "Soft error rate mitigation techniques for modern microcircuits," 2002 IEEE International Reliability Physics Symposium. Proceedings. 40th Annual (Cat. No.02CH37320), Apr. 2002, pp. 216-225.

[94] M. Berg, “Single Event Effects in FPGA Devices 2014-2015”, NASA Electronic Parts and Packaging Program (NEPP) Electronics Technology Workshop (ETW), June 2015.

[145] K. S. Morgan, D. L. McMurtrey, B. H. Pratt and M. J. Wirthlin, "A Comparison of TMR With Alternative Fault-Tolerant Design Techniques for FPGAs," IEEE Trans. on Nuclear Science, vol. 54, no. 6, pp. 2065-2072, Dec. 2007.

[174] F. G. de Lima Kastensmidt, G. Neuberger, R. F. Hentschke, L. Carro and R. Reis, "Designing fault-tolerant techniques for SRAM-based FPGAs," IEEE Design & Test of Computers, vol. 21, no. 6, pp. 552-562, Nov.-Dec. 2004.

[175] B. Pratt, M. Caffrey, J. F. Carroll, P. Graham, K. Morgan and M. Wirthlin, “Fine-Grain SEU Mitigation for FPGAs Using Partial TMR”, IEEE Trans. on Nuclear Science, vol. 55, no. 4, pp. 2274–2280, Aug. 2008.

[163] A. J. Sanchez-Clemente, L. Entrena, and M. Garcia-Valderas, “Partial TMR in FPGAs using approximate logic circuits”, IEEE Trans. on Nuclear Science, vol. 63, no. 4, pp. 2233–2240, Aug. 2016.

[162] A. J. Sánchez Clemente, L. Entrena y F. Kastensmidt, "Approximate TMR for selective error mitigation in FPGAs based on testability analysis," 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Aug. 2018.

[97] B. Shim and N. R. Shanbhag, "Reduced Precision Redundancy for Low-power Digital Filtering," Proc. 35th Asilomar Conference on Signals, Systems and Computers, pp. 148-152 vol.1, Nov. 2001.

[98] B. Shim and N. R. Shanbhag, "Energy-Efficient Soft Error-Tolerant Digital Signal Processing," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 4, pp. 336-347, Apr. 2006.

[170] S. Liu, K. Chen, P. Reviriego, W. Liu, A. Louri and F. Lombardi, "Reduced Precision Redundancy for Reliable Processing of Data," IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 4, pp. 1960-1971, 1 Oct.-Dec. 2021

[168] B. Pratt, M. Fuller and M. Wirthlin, "Reduced-Precision Redundancy on FPGAs," Int. Journal of Reconfigurable Computing, vol. 2011, Article ID 897189, Oct. 2011.

[105] W. Liu, Q. Liao, F. Qiao, W. Xia, C. Wang and F. Lombardi, "Approximate Designs for Fast Fourier Transform (FFT) With Application to Speech Recognition," IEEE Trans. on Circuits and Systems I: Regular Papers, vol. 66, no. 12, pp. 4727-4739, Dec. 2019.

[8] L.A. García-Astudillo, A. Lindoso, M. Portela and L. Entrena, "Evaluation of a Reduced Precision Redundancy FFT Design," XXXV Conference on Design of Circuits and Integrated Systems (DCIS), Nov. 2020.

[74] Xilinx Inc., “Soft error mitigation controller v4.1 Product guide,” White Paper PG036, Nov. 2014.

[157] E.H. Wold and A.M. Despain. “Pipeline and parallel-pipeline FFT processors for VLSI implementation”. IEEE Trans. Comput., C-33(5), pp. 414-426, May 1984.

[1] L. A. García-Astudillo, A. Lindoso, L. Entrena, H. Martín, P. Martin-Holgado, M. García-Valderas. “Analyzing Reduced Precision Triple Modular Redundancy Under Proton Irradiation”. IEEE Nuclear & Space Radiation Effects Conf. (NSREC), July 2021 (submitted to IEEE Trans. on Nuclear Science).

6. REDUCED RESOLUTION REDUNDANCY: A NOVEL APPROXIMATE ERROR MITIGATION TECHNIQUE

The interesting results found in the Reduced Precision Redundancy encouraged us to lead our research towards the development of new, unexplored Approximate Error Mitigation techniques. The Reduced Resolution Redundancy technique, presented in this chapter, is the first of the original contributions carried out during this Thesis. This technique leverages the flexibility of hardware-based implementations of Digital Signal Processing algorithms to reduce their consumption of resources in exchange for longer computation times. However, by means of approximate calculations based on the reduction of resolution in the input data, the computation time can also be reduced.

We demonstrated this novel approach using fault injection on a Fast Fourier Transform benchmark with an ad-hoc architecture to satisfy our needs.

This chapter has been published as an article:

[4] © 2022 IEEE. Reprinted, with permission, from L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, *et al.*, "Reduced Resolution Redundancy: A Novel Approximate Error Mitigation Technique," *IEEE Access*, vol. 10, pp. 20 643–20 651, 2022. DOI: 10.1109/ACCESS.2022.3152202

Abstract

Error mitigation techniques, such as Triple Modular Redundancy, introduce very large overheads. To alleviate this overhead, approximate techniques can be used. In this work we propose a novel Approximate Error Mitigation technique based on using redundant circuits with lower resolution. As a representative case study, the approach is demonstrated for a Fast Fourier Transform, for which an optimized architecture is proposed. The approach is validated through fault injection. Experimental results show that Reduced Resolution Redundancy can significantly reduce the overhead and achieve an excellent error mitigation performance and a low sensitivity to uncorrectable errors.

6.1. Introduction

Soft errors are becoming a concern for an increasing number of applications. In the past, they were mainly relevant in high reliability applications working in harsh environments, such as space. With the shrinking of technologies and the increase of complexity and density, the susceptibility to soft errors has grown to be significant for a large variety of applications, including many at the ground level such as communications, autonomous vehicles, medical appliances, and high-performance computing.

The most important cause of soft errors is ionizing radiation [176]. There are radiation-hardened technologies that can reduce the susceptibility to soft errors. However, these technologies are very expensive and lag way behind commercial technologies. As a matter of fact, Commercial Off-The-Shelf (COTS) electronic components are currently being considered even for space applications [177], due to their higher performance, lower power consumption, higher availability, and lower cost. While in the past reliability was a major obstacle for the adoption of COTS in safety-critical applications, today the balance is more in favor of taking advantage of COTS benefits and use error mitigation techniques to cope with soft errors. For less critical applications, where dependability is not so stringent, or for low error rates, such as those found at the ground level, systems built on commercial devices are thought to be a good alternative to radiation-hardened systems.

COTS generally do not provide error mitigation capabilities, which must be implemented by the designer. Exceptionally, some components include some partial protections. For instance, on-chip memories increasingly support Error Detection And Correction (EDAC) codes, communication interfaces include parity bits or EDAC codes, etc. However, the logic is generally not protected. A very popular and general hardware error mitigation technique is Triple Modular Redundancy (TMR) [178]. TMR is quite effective, but it also needs more than thrice the resources of the original system. For this reason, the development of alternative mitigation techniques able to correct errors and keep a moderate consumption of resources at the same time, is very enticing.

Approximate error mitigation techniques aim at reducing the usage of resources by implementing approximate versions of the design instead of full copies for the redundant modules. Several Approximate Triple Modular Redundancy (ATMR) techniques have been proposed for general logic circuits [160]-[161]. For Digital Signal Processing (DSP) designs, Reduced Precision Redundancy (RPR) [97]-[105] is another example of Approximate Error Mitigation methods. RPR uses redundant modules with reduced precision that produce an approximate result. If needed, a carefully tailored voting logic may be used, which is capable of detecting and correcting errors with a tolerable accuracy.

In this paper we present a novel Approximate Error Mitigation technique based on the addition of two redundant modules that implement reduced resolution versions of the design to protect. The approach is general enough to be applied to a wide variety of algorithms. As a case study, we illustrate this approach with a Fast Fourier Transform (FFT) design, which is a representative example of a DSP algorithm. In the proposed Reduced Resolution Redundancy technique, the redundant modules calculate frames with less points. As a result, the redundant modules can complete the computation in less time or with less resources. Based on this premise, an optimized FFT architecture is proposed which optimizes the use of resources while it adjusts to the timing requirements of the full resolution module. Our results show that this technique achieves an excellent error mitigation performance and a low sensitivity to uncorrectable errors.

The rest of the paper is as follows. Section 6.2 summarizes related work on

Approximate Error Mitigation techniques. Section 6.3 describes the proposed Reduced Resolution Redundancy approach. Then, in section 6.4, this approach is applied to the case of an FFT design. In Section 6.5 we present the experimental setup used in this work. Section 6.6 discusses the experimental results. Finally, section 6.7 shows the conclusions of this work.

6.2. Background and related work

Error mitigation techniques use redundancy in order to be able to differentiate correct behavior from wrong behavior. Hardware redundancy uses redundant copies of the design to detect or correct errors [178]. Duplication With Comparison (DWC) and Triple Modular Redundancy (TMR) are the most commonly used solutions to detect and correct errors, respectively. These solutions are quite effective, but they involve a very high overhead. Ideally, TMR can correct any error that appears in one of the three modules, but it requires more than three times the logic. In low cost or low power cases, this large overhead may not be acceptable. To alleviate this overhead, some authors have studied partial error mitigation solutions that perform a selection of the components to replicate [166]. Alternatively, approximate redundant solutions have been proposed.

The Approximate TMR (ATMR) approach may not be able to correct some errors but can reduce the area and power overhead. The goal is to find an optimal trade-off between the error probability and the hardware overhead. A general approach for logic circuits is proposed in [160] using synthesis techniques. The approximate redundant circuits are simplified circuits that perform a different, but closely related logic function, so that they can be used for error detection [160] or error masking where they overlap with the original circuit [179], [180]. The approximations are performed by selectively simplifying the parts of the logic that are less relevant from the functional point of view. Probabilistic analysis is used to determine the most suitable logic transformations [180]. In [179], approximations are generated by substituting the least testable lines by logic constants. The rationale of this approximation approach is that errors in lines with low testability have few chances of being propagated. This approach is extended and compared with evolutionary techniques in [161].

Reduced Precision Redundancy is an ATMR technique that has been proposed for DSP circuits, [97]. Fig. 6.1 shows a block diagram of the RPR technique for an FFT design. Two redundant Reduced Precision (RP) modules are used along with a Full Precision (FP) module. The RP result approximates the FP result with a certain error threshold. If the two RP modules produce the same result and the difference with the FP result exceeds the threshold, then the FP result is corrected by the RP result. Otherwise, the FP result is considered correct.

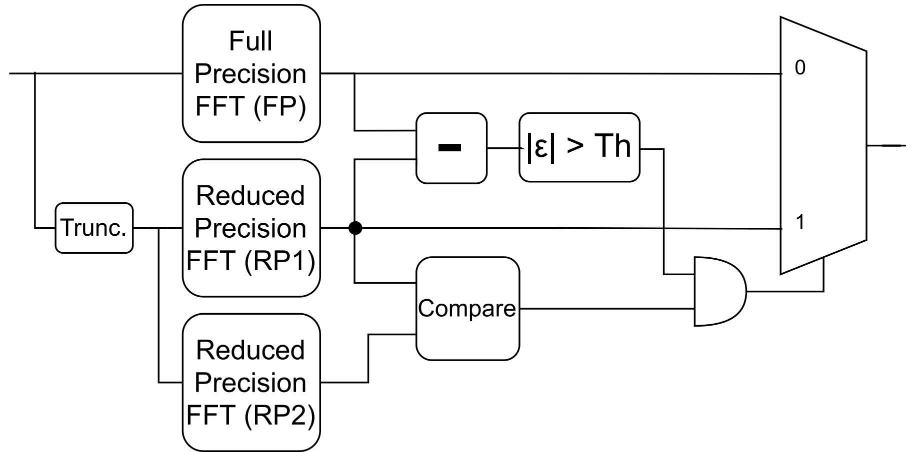


Figure 6.1: Block diagram of a Reduced Precision Redundancy system to correct errors in FFT blocks.

RPR was first proposed as a means to enable aggressive voltage scaling to reduce power consumption and compensate for the possible errors produced by the delay increase [97], [98]. This technique was analyzed by fault injection in [168] for the case of FIR filters. The results show that RPR can produce good error mitigation at a fraction of the area overhead of TMR. In [1], the authors validate it using radiation experiments for a Fast Fourier Transform (FFT).

6.3. Reduced Resolution Redundancy

In this section we describe the Reduced Resolution Redundancy (RRR) proposal. As in ATMR and RPR approaches, RRR is based on the addition of two redundant versions to the design to protect. Each redundant version only calculates an approximation to the results of the original design, aiming at reducing the resources used by the redundant modules when compared to a conventional exact TMR approach.

In many structured algorithms, resolution is a key parameter, which has a large impact on the computational effort. Thus, the error mitigation overhead can be reduced by using lower resolution in the redundant modules. For example, the redundant copies of an RRR image processor would use lower resolution pictures, i.e., pictures with reduced size. By contrast, an RPR image processor would use pictures of the same size, but with less bits per pixel. If we apply this concept to the FFT algorithm, which is the case study used in this work, the redundant copies would need to calculate FFT frames of reduced length.

Reduced resolution processing requires less effort, resulting in less time, less resources or both. In most cases, time and resources can be interchanged by using appropriate architectures. In hardware redundancy, the goal is to reduce the hardware size while keeping the throughput, so that the overall computation time is not penalized. For instance, a lower resolution image reduces the processing time if the same image processor is used. Alternatively, the image processor can be simplified to save resources, thus increasing the processing time up to matching that of the full resolution image.

In many DSP architectures it is common to use multiple processing elements and processing stages. In case the processing elements are all identical, the application of the proposed RRR technique is immediate. Let N be the amount of data to be processed and M be the amount of processing elements, so the processing time should be O(N/M). If the reduced resolution copies reduce the resolution by a factor of k, the amount of data to be processed becomes N/k and we can compensate it by reducing the amount of processing elements by the same factor k. Thus, the RRR copies can reduce the size by k while meeting the required processing time. In contrast, in RPR, the redundant copies use the same amount of processing elements, but each element is smaller. In the cases where the precision of the processing elements is fixed, RPR is not beneficial, but RRR can be useful.

In the general case, when the processing elements are not identical, the application of the RRR techniques may not be so evident. However, it is generally possible to trade off time for resources. This is a common task for digital designers and is also supported by high-level synthesis tools. The goal is to use a lower resolution for the redundant modules and then reduce their size as much as possible within the time constraints of the full resolution design.

Redundancy and voting approach

As in the case of other error mitigation techniques, RRR technique is comprised not only by the design to protect and its redundant copies, but also by a voting logic circuit capable of correcting errors found in the results of either copy. In the case of RRR, there is the additional problem of comparing and voting data sets of different sizes. In order to overcome this problem, a conditioning step must be performed at the output of the Reduced Resolution (RR) modules before they can be compared with the Full Resolution (FR) result in the voting logic.

Fig. 6.2 shows the basic architecture of a fault-tolerant design using the RRR technique. The inputs of the three modules should represent the same data, but the RR modules should receive less data points. It is assumed that the results with reduced resolution can be matched in some way with those of higher resolution. In general, the way of matching RR results may depend on the particular application.

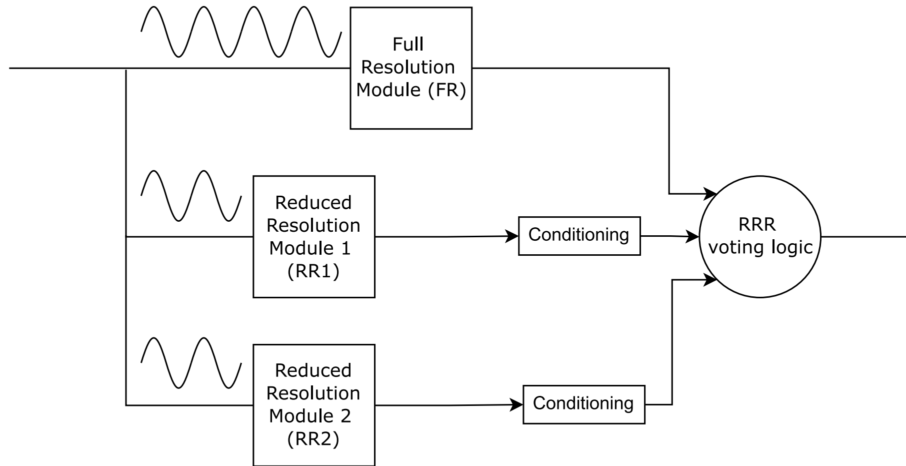


Figure 6.2: Block diagram of the proposed Reduced Resolution Redundancy technique.

The conditioning aims to establish a one-to-one relation between the result points of the FR and the RR modules. Several scaling algorithms can be adopted for this task. The simplest approach, and the one used in our case study, is the so called nearest-neighbor interpolation: the gaps caused by expanding the image are filled with the values of the nearest neighbor points. This approach may be complemented by ad-hoc techniques intended to correct some particular types of errors.

The proposed scheme relies on expanding the RR results so that they match the size of the FR results. As a consequence, some noise may be introduced on the RR data, but it can nevertheless be used to detect and eventually correct errors in the FR data.

6.4. Application to FFT case study

The case study used in this work is an FFT architecture, namely a pipelined FFT architecture. This is a complex design that includes adders, multipliers, and FIFO memories in a pipeline, with relevant synchronization constraints. The pipeline stages are all different, so the application of the RRR technique is not trivial.

Among the various existing FFT pipelined architectures, we selected the Radix-2 Single-path Delay Feedback (R2SDF) model [157]. Other architectures with higher radix can be used alike and the proposed approach can be applied similarly to other designs.

To take advantage of RRR, we developed a novel FFT architecture to be used for the redundant modules of the mitigated R2SDF FFT design. We call this architecture R2SER2SDF (Radix-2 Serial-2 SDF). The goal of the R2SER2SDF design is to optimize the resources under the constraint that the processing time for a half-resolution FFT frame matches that of the full resolution FFT frame. It is aimed at reducing the resources and increase the latency by serializing stages. To this purpose, each stage implements two half-resolution stages in a serial manner.

The R2SER2SDF architecture

Fig. 6.3 shows the construction of a Radix-2 SDF FFT pipelined architecture. It is comprised of log2N stages, where N represents the number of data points in the FFT calculation. Each stage contains a Butterfly Unit (BU), that computes sums of complex numbers, a FIFO memory to delay some of the data that serve as input for the BU and a multiplier unit that performs complex multiplications of the BU's outputs by the constant twiddle factors. In each stage, the BUs are identical, but the size of the delay buffers is divided by a factor of two. In each clock cycle, each stage processes one input point and produces an output point after some latency. Stages work in pipeline, so they can process several FFT frames in a seamless manner.

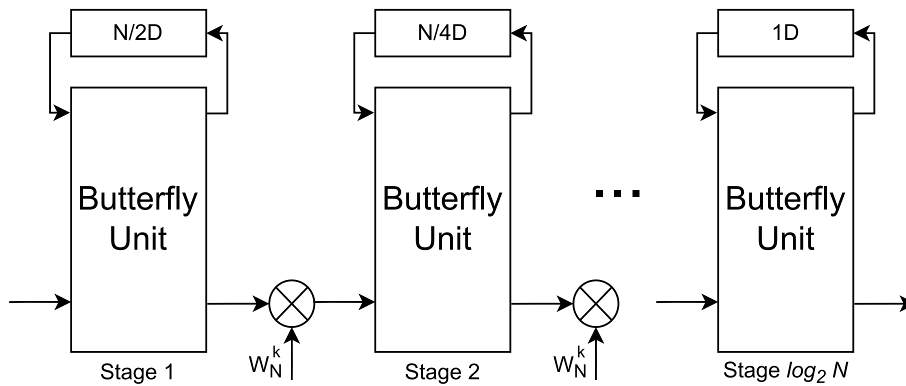


Figure 6.3: Architecture of a Radix-2 SDF pipelined FFT.

In Fig. 6.4 we can see the proposed R2SER2SDF architecture. This implementation uses log4N stages, half the stages needed in the R2SDF architecture, but it is still a radix-2 architecture. This is accomplished by merging two adjacent stages in such a way that they are carried out using a single BU and a single multiplier. Each stage in the new architecture performs two rounds of computations in a serial manner. If used for half size data frames, it can complete the two rounds of computations in the same time a R2SDF stage processes a full-size data frame.

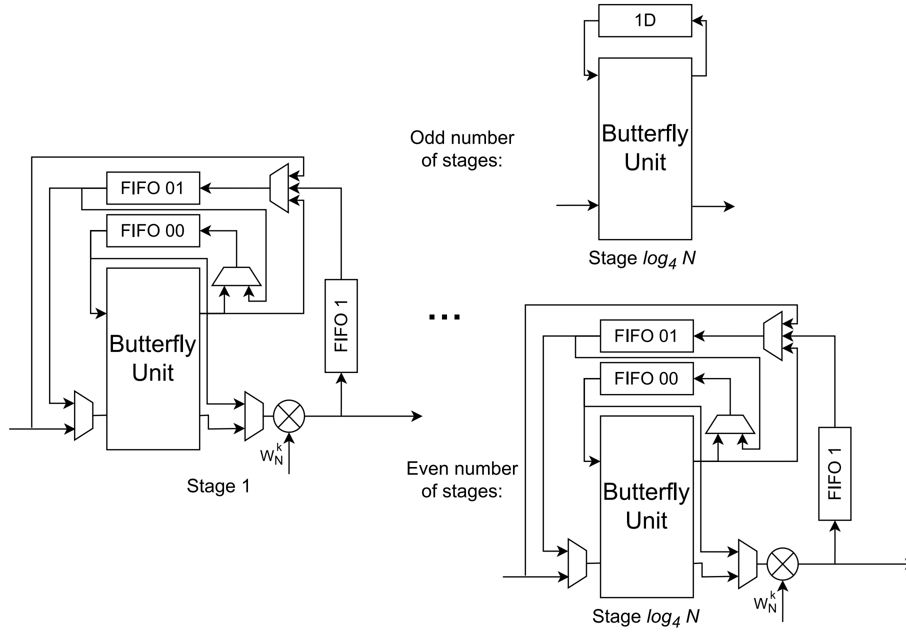


Figure 6.4: Architecture of the proposed R2SER2SDF pipelined FFT.

The delay FIFO buffer is divided in two halves. In the second round, only the second half is used. Finally, another FIFO (FIFO1) is added to delay the result of the first round of calculations, so the result from the first calculation is stored in a FIFO memory and fed back to the input of the stage to perform the second calculation, after which the resulting data are passed to the next stage. Since this method requires pairing the Radix-2 SDF stages, it can only be fully applied when log2N is even. In case it is an odd number, the last stage of the pipeline has to be instantiated as a regular R2SDF stage.

Fig. 6.5 shows FPGA synthesis results, namely the number of Flip-Flops (FF), Look-Up Tables (LUTs) and Digital Signal Processing (DSP) blocks, for a variety of Radix-2 pipeline FFT implementations using different precisions and resolutions. These FFTs will be used later on as part of the TMR, RPR and RRR error mitigation architectures, to protect a Full Precision FFT with two other instances of Full Precision (TMR), Reduced Precision (RPR) or Reduced Resolution (RRR) modules, respectively. The first group of designs in the graph shows synthesis results for Reduced Precision FFTs (RP), that use half the precision of the Full Precision FFTs (FP) shown in the second group of the table. The RP modules have been used in previous studies to implement Reduced Precision Redundancy error mitigation [1]. The last group displays synthesis results for the Reduced Resolution FFTs (RR) implemented with the R2SER2SDF architecture for half resolution. All the results are for 256-point frames, except in the RR case, which uses 128-point frames. The designs were synthesized for a Xilinx Zynq-7010 device.

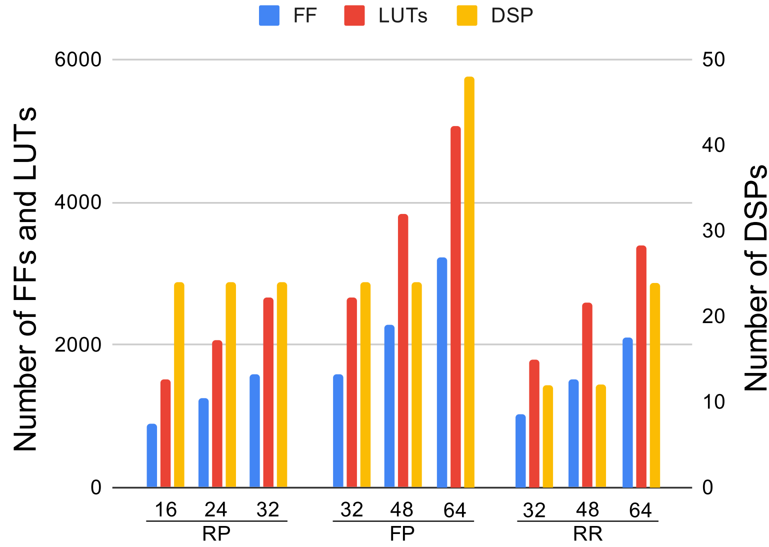


Figure 6.5: Utilization of resources of single FFTs with different precisions and resolutions, to be used in the TMR, RPR, and RRR error mitigation techniques.

From the data in the graph, we can conclude that the RP and RR FFTs effectively reduce the resources needed for the implementation, compared to an FP FFT. We can also see that the RR FFT at low precisions uses less DSP blocks, while the amount of FFs and LUTs are slightly higher than the RP FFT due to the need of an extra FIFO in each stage. When using 64 bits of precision, the DSP blocks increase in the RR FFT. This can be easily explained by the size of the DSP multipliers in the Zynq-7010. One of these DSPs can accommodate the multiplication of the real or imaginary parts of the data by the twiddle factors of the FFT if the data word is not greater than 25 bits. This way, for the 32- and 48-bit FFTs, which use 16- and 24-bit wide real and imaginary parts, each multiplication can be performed by a single DSP, while in the 64-bit FFT two DSPs are needed. The Reduced Precision blocks of the RPR still benefit from this effect in the 64-bit RPR and that is why the RPR uses less DSPs than the FP FFT in this case. In contrast, the RR FFT always uses half of the DSPs of the FP FFT.

Signal conditioning

The results of the RR copies can significantly differ from the results of the FR copy, and they need to be carefully treated before the copies can be compared to detect and correct errors. Since the FFT can be considered as a one-row image, the nearest neighbor of a gap is the previous point. For the chosen implementation, in which the RRs have half the points of the FR, the nearest neighbor interpolation can be easily performed by holding the RR output value two clock cycles instead of the one clock cycle originally needed. However, this interpolation introduces some distortions. First, the complex values of the outputs are quite different between the FR and the RR. Instead of comparing the complex values, as a typical RPR voter would do, the proposed RRR voting logic uses the modules

of the complex numbers to perform the comparison. Second, the frequency peaks in the RRs are now twice as wide as the same peaks in the FR and this shall be taken into account in the voting logic. These two issues introduced by the nearest-neighbor interpolation can be seen in the frequency response shown in Fig. 6.6. The comparison of the FR and RR results taking these issues into account is described in the following section.

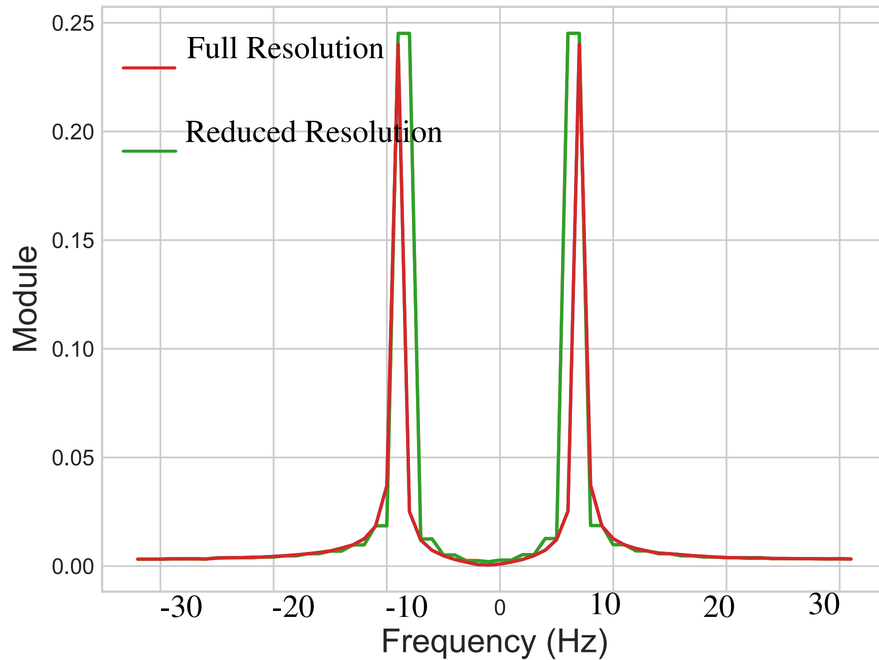


Figure 6.6: Comparison of the outputs of the Full-Resolution FFT and the Reduced-Resolution FFT after the conditioning step.

Voting logic

The main idea behind the voting logic is similar to that of an RPR voter. It is based in the following principles:

- The FR output will be considered correct unless the RRs outputs suggest otherwise.
- Since the FR and the RR values are not exactly the same, an acceptable error, in the form of a threshold value, is considered. If the RR values are identical and the difference between the FR and the RR values exceeds the threshold, the FR value is considered incorrect and the RR value is taken as the best available approximation to the correct value.
- If the RRs outputs are not identical, their values cannot be trusted, and the FR value will be used as the voted output.

In addition, to overcome the inaccuracies introduced in the interpolation of the RR results, two other rules are considered. First, the comparison of the FR and RR

results is carried out using the module of each FFT point. For simplicity, we use a pseudo-module obtained as the sum of the absolute values of the real and imaginary parts of the complex data, whose value should be similar enough to the actual module. Nevertheless, the correction of errors in the FR will use the complex value of the RRs, not the pseudo-module. Second, since the frequency peaks in the RRs are wider than in the FR due to the interpolation, the voter must check the modules not only in the incoming data point, but also in the previous one, to ensure errors and correct results are classified accurately.

The voting algorithm that includes all these rules is summarized in Fig. 6.7. A block diagram of the output selection algorithm is shown in Fig. 6.8.

RRR Voting algorithm

Inputs:
FR module output
RR modules outputs

Output:
Voted result

Process:

1 Calculate pseudo-module of FR and RR1.

2 Calculate RR1 == RR2.

3 Detect frequency peak in FR module.

4 Detect frequency peak in RR1 module.

5 Calculate if the difference between the modules is greater than a defined threshold.

6 **if** a peak was detected in FR in the past clock cycle and both RR are equal in this clock cycle and the previous one and the error was greater than the threshold **then**, correct the result with RR1.

7 **else if** no peak was detected in FR in the last three clock cycles and both RRs are equal in this clock cycle and the previous one and there was a peak in the RRs **then**, correct the result with RR1.

8 **else**, the FR result is correct, the FR has an error smaller than the threshold or the RR1 and RR2 are different. **Then**, do not correct with RR1.

Figure 6.7: Proposed RRR voting algorithm for error correction.

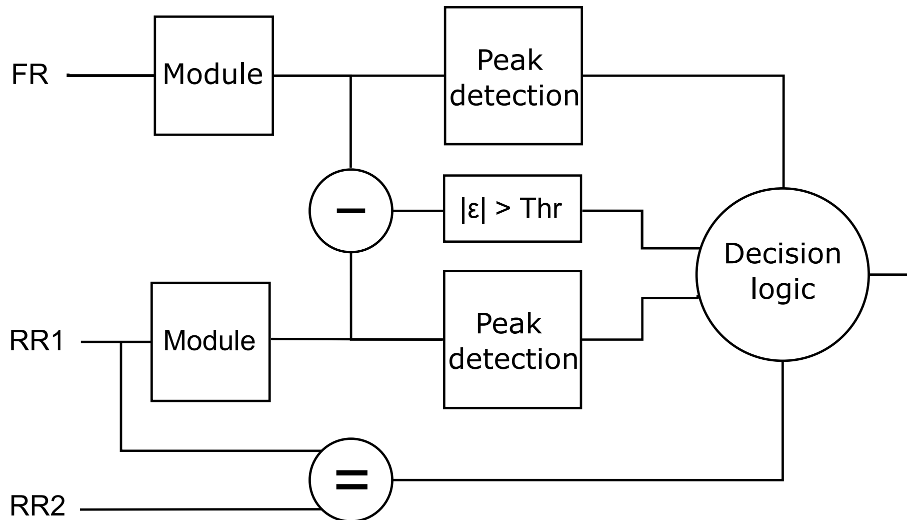


Figure 6.8: Structure of the proposed voting algorithm for the RRR hardening technique.

This algorithm is capable of correcting three types of errors, examples of which can be seen in Fig. 6.9. First, if an error greater than the threshold appears in a frequency peak in the FR, that value will be corrected with the RR data point (this corrects unexpected peaks and peaks with slightly incorrect values, explained in Step 6 of the algorithm), as seen on Fig. 6.9a. Second, frequency peaks found in both RR modules but not in the FR module can be corrected in the voted output. This kind of error is shown on Fig. 6.9b and explained in Step 7 of the algorithm. Since peaks in the RRs are wider, when correcting errors this way, the peak in the correct output is also wider, which means the voted output has less precision in the detection of frequencies, but it is nevertheless an acceptable approximate result. Third, errors in one of the RR modules, as the ones shown in Fig. 6.9c, will be effectively masked (Step 9 of the algorithm). Finally, as in other single error mitigation techniques, errors affecting more than one module at the same time may affect the voted output in unpredictable ways, as shown in Fig. 6.9d. The latter produces a Common Mode Failure (CMF), which is of the outmost importance in FPGAs, because there are configuration bits that can have effects in more than one redundant domain [146].

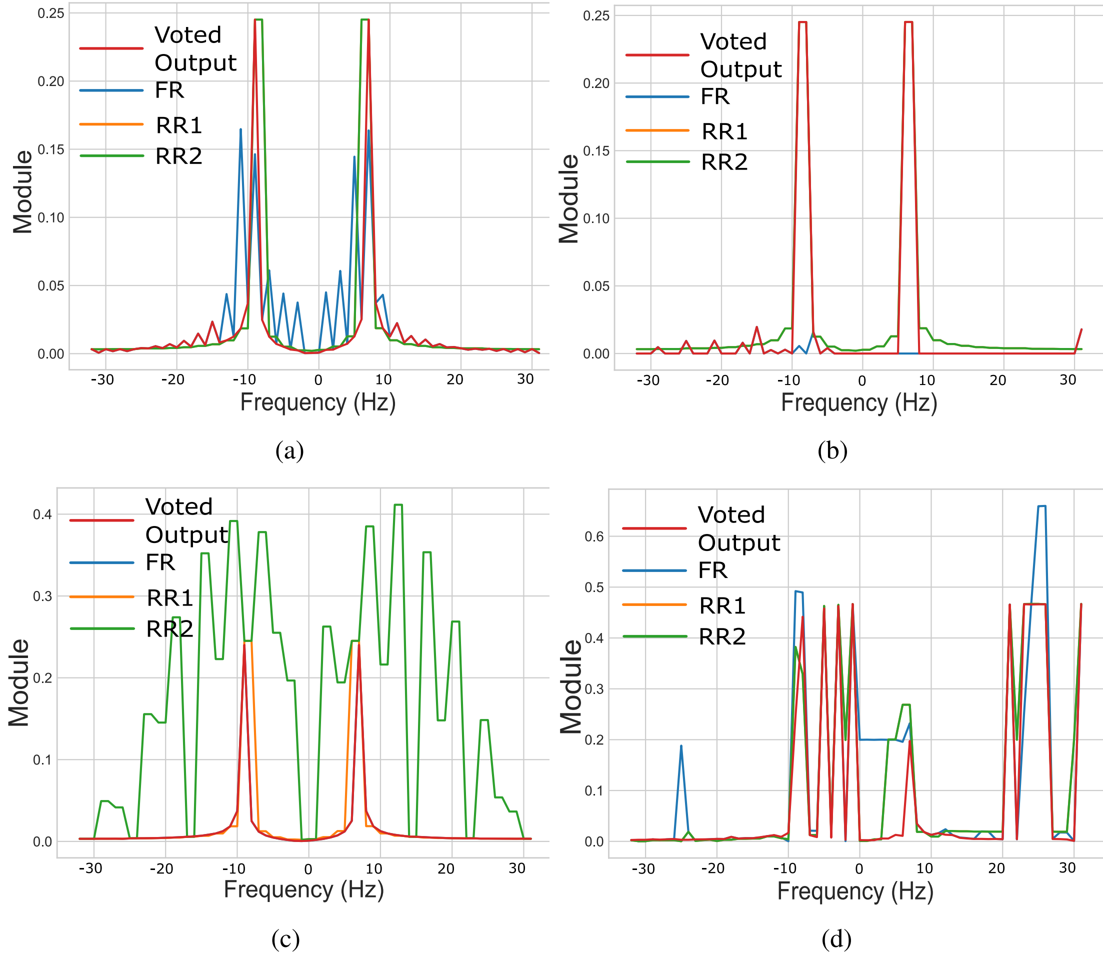


Figure 6.9: Behavior of the voting algorithm when exposed to four possible types of errors in the FFTs: (a) correction of FR peak error; (b) correction of FR peak absence; (c) masking of RR error and (d) Common Mode Failure.

It is worth noting that because the voting process relies on the detection of peaks in the module of the FFT frame, this algorithm seems to be tailored for the FFT case. However, this solution can be easily adapted to harden other circuits. The peak detection is nothing but a comparison against a threshold and it is only used to include frequency peaks in the voted output when the FR module failed to compute them. This is a measure to ensure that the most important information of the FFT is not completely lost due to a fault. The idea of peak detection could also be useful in algorithms such as edge detectors and the sensitivity of the detection can be adjusted depending on the application. Setting this threshold level to 0 completely disables this feature, transforming the RRR voter in a slight modification of an RPR voter that can be used in any algorithm other than the FFT.

6.5. Experimental setup

The performance of the RRR error mitigation technique has been assessed with fault injection experiments. We developed an RRR system to harden a Radix-2 SDF pipelined

FFT, adding two copies of the proposed R2SER2SDF architecture as the redundant modules. The design comprises a R2SDF FFT with a frame length of 256 data-points and two R2SER2SDF 128-point FFTs. To cope with possible errors in the voting logic, we have used triplicated voters, in a Distributed TMR fashion [94].

Because the RRR voter is more complex than the typical RPR or TMR voters, it would be expected to be even more sensitive by its own, and triplicating it could even be counterproductive, making the voter more prone to CMFs. We decided to create two benchmarks based on the same RRR system, one with a single voter and another one with triplicated voters, and test whether triplicating the voters is worth the additional area usage in this case.

The RRR system was wrapped in a testbench capable of controlling the inputs and outputs of the FFTs and checking their correctness. This testbench stores the resulting frames of the FFTs and the voted output, and compares them against the expected results. Whenever an error is detected in any of these frames, they are sent through a UART connection to an external controller for further offline analysis and the FPGA is immediately reset and reconfigured to prevent error accumulation.

We implemented this setup in the programmable logic of a Xilinx Zynq-7010 All Programmable SoC (APSoC) [138]. This device also has a dual core ARM processor, but we did not use it in our designs.

The injection experiments were performed using Xilinx Soft Error Mitigation IP [74], which is used to scrub the configuration memory of the FPGA and inject faults in random addresses of the memory to analyze the error correction capabilities of the design.

6.6. Results

In this section we detail the synthesis results, and the outcome of the fault injection experiments of the proposed RRR design for the FFT case study. We compare the results with TMR and RPR error mitigation designs that we tested using the same setup.

The FFTs were configured to calculate 256-point frames with 32-bit precision (16-bit real part and 16-bit imaginary part) and the redundant copies in the corresponding RPR and RRR implementations introduce modifications on these parameters to save resources. The Reduced Precision copies of the RPR design use a precision of 16 bits. The Reduced Resolution copies of the RRR designs compute 128-point frames, with 32-bit precision.

We have implemented two versions of the RRR error mitigation architecture, one with triplicated voters and one with a single voter, in order to explore the impact on the number of uncorrectable errors due to the triplication of voters and study if it is worth the extra resources it needs.

Synthesis

Fig. 6.10 presents the usage of Flip-Flops (FFs), LUTs and DSPs of several implementations. The first group of results on the left compare the TMR, RPR and RRR implementations for the nominal 32-bit precision. When comparing the usage of resources of the RRR design we can make two observations. First, the RRR design actually achieves a significant area reduction with respect to TMR, although the RPR design is still more efficient in these terms. On the contrary, the RRR designs only use 48 DSP blocks, while both TMR and RPR designs use 72 DSPs. Second, the triplication of the voting logic has a large impact on the usage of LUTs as logic. This is not shown in the figure, which only includes the results using triple voters. The RRR design with a single voter has 12% less LUTs than the design with triple voters, getting close to the usage of LUTs as logic of the RPR design.

We also extended the analysis to higher precision FFTs. The second and third groups of columns in Fig. 6.10 show the utilization of FFs, LUTs and DSP blocks of the TMR, RPR and RRR designs when using 48-bit and 64-bit data words. There is a linear trend between the number of LUTs and FFs, and the precision used for the calculations. However, the number of DSP blocks used to perform the multiplications varies significantly when 64-bit words are used, because of the size of the DSP multipliers. Multiplications of real or imaginary parts of data up to 25 bits by the twiddle factor in an FFT stage can be performed by a single DSP slice in the Zynq-7010 device. Thus, multiplications in 48-bit FFTs (24-bit real and imaginary parts) can still be accommodated in a single DSP slice. For higher precisions, namely 64-bit FFTs, each multiplication needs two DSP slices.

We can expect that error incidence will increase for the designs with 48-bit and 64-bit widths because of the greater usage of resources, but the distribution of errors should be similar. The non-linear increase of DSP blocks may cause a slight increase in the incidence of errors in the 64-bit width, besides the predictable increase due to the increase of FFs and LUTs. The experimental results shown in this article all refer to 32-bit FFTs, but they could be reasonably used as a reference for higher precisions based on the above considerations.

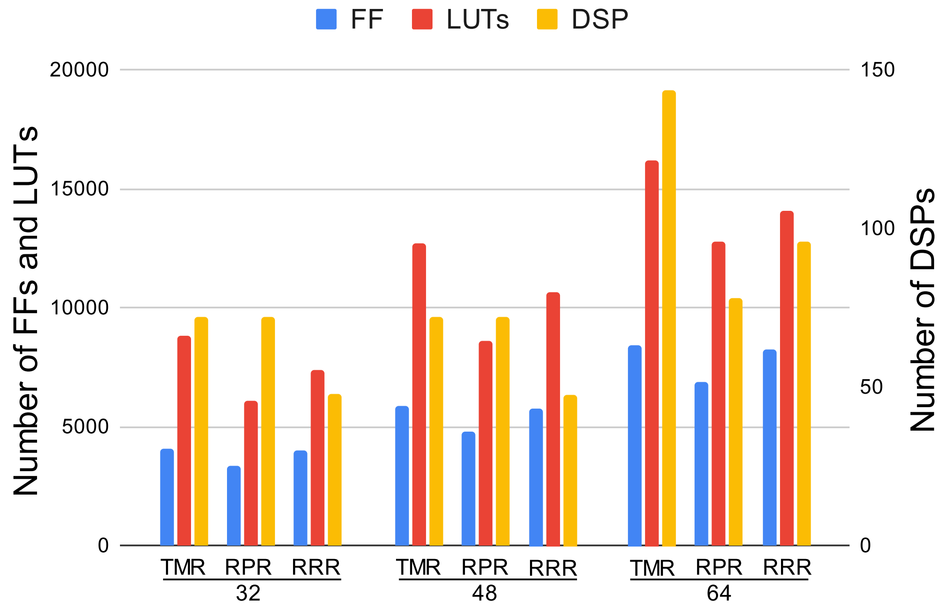


Figure 6.10: Utilization of resources of the different designs for increasing data precisions.

Fault Injection experiments

To evaluate the error mitigation performance of the RRR design, we conducted a series of fault injection experiments injecting faults in the configuration memory of the FPGA and collecting the erroneous frames produced in the FFTs. We performed an offline statistical analysis of the collected data to classify the errors and obtain information about the sensitivity of the system and its failure modes. Tables 6.1 and 6.2 summarize the main metrics we extracted in the analysis. Table 6.1 shows information related to the number of faulty frames found. In the first row we show the number of injected addresses in the configuration memory during the experiments. The next two rows show the number of faulty FFT calculations and the error rate, computed as the number of faulty frames divided by the number of injections performed. This metric offers a measure of the sensitivity to failure of the benchmark, and we can use it as a comparison method. Between parenthesis we have included the confidence intervals for the number of faulty frames using a Gaussian estimation at 95% confidence. The next two rows report the amount of frames with errors that cannot be corrected using the presented hardening techniques. Frames with errors affecting the result of more than one FFT at the same time are classified as Common-Mode Failures (CMF). Frames in which the error appears in the output of the voting logic are classified as voting logic errors. This kind of errors can either affect the voting logic or some point between the voters and the logic that checks the correctness of the results. The error rate for these two types of uncorrectable faults is calculated in the next row using the same method previously described. In the last row of the Table, we calculated the average Peak Signal-To-Noise Ratio (PSNR) of the faulty frames for each benchmark. This number, calculated as in Eq. 6.1, measures

the precision of a noisy signal. A higher result indicates less distortion.

$$PSNR = 10 \times log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{6.1}$$

Table 6.1: RESULTS FROM THE INJECTION EXPERIMENTS

	TMR (Triple voter)	RPR (Triple voter)	RRR (Triple voter)	RRR (Single voter)
Injected addresses	360000	360000	360000	360000
Faulty frames	1649 (1569, 1729)	1423 (1349 ,1497)	1352 (1280, 1424)	1506 (1430, 1582)
Error rate (10^{-3})	4.62	3.95	3.76	4.18
Frames with CMF	51	46	8	9
Frames with voting logic errors	0	10	6	14
Uncorrectable errors rate (10^{-5})	14.28	18.48	3.92	6.44
PSNR (dB)	77.3	79.9	65.76	68.8

From the data in Table 6.1, we can clearly appreciate a reduction in the error rate of the RRR design, associated with a decrease of the number of faulty frames, compared to the TMR and RPR designs. This behavior can be explained by the significative reduction of resources, specially LUTs and DSPs, achieved by the RRR design. The RRR design using a single voter has a higher error rate than the one with triple voters. This increase was expected, since the triplication of the voting logic would mitigate some errors, lowering the error rate.

More importantly, both RRR designs have much better results in CMFs and voting logic errors according to the data in Table 6.1. This result demonstrates that the RRR technique has a superior error correction capability. Even without triplicating the voting logic, the rate of uncorrectable errors is significantly lower than that of TMR and RPR. The triplication of the voting logic is definitely worth the resources, since uncorrectable errors in the voting logic have been greatly reduced by the TMR hardening in the voting logic.

The PSNR, calculated with the voted result of each frame in the RPR and RRR designs, is better in the RPR case, since the data used to correct faults has better precision; the expanded results from the Reduced Resolution modules are approximations, not only because the numerical results are slightly different to the FR, but also because expanding the frame to the full length can also introduce errors in the corrected results. The PSNR of the TMR design was calculated at the output of the FFT modules, not at the output of the voter, because only CMF and voting errors reach the output of the voter and the PSNR value would be biased and useless to establish a comparison. A more detailed analysis of the faulty frames is provided in Table 6.2. Each FFT frame of our 256-point FFTs contains 512 words. These words correspond to the real or imaginary parts of each complex data point. For the basic FFTs, each word is 16-bit wide, but the Reduced Precision FFTs

modify this length in order to reduce the resources needed in the computations. In the table we have calculated the total amount of words in the faulty frames we received from the FPGA and how many of those were actually erroneous, as seen on the first two rows of the Table. Then, we classified the erroneous words according to the impact they had on the voted output by comparing the results of the three FFTs and the voted output in the corresponding data point. We created two main groups. Tolerable errors include errors that were masked and did not reach the voted output, errors that were totally or partially corrected and errors that did not exceed the tolerable threshold in the RPR and RRR designs. The category CMF and voting logic errors comprises errors that reached the voted output by affecting more than one FFT at the same time or directly affected the voted result. These are also referred to as Uncorrectable errors.

Table 6.2: ERROR CLASSIFICATION OF THE FAULTY WORDS IN THE ERRONEOUS FRAMES

	TMR	RPR	RRR (triple voter)	RRR (single voter)
Total words in faulty frames	844288	728576	692224	771072
Faulty words	267648 (31.7%)	133201 (18.3%)	155939 (22.5%)	158076 (20.5%)
Tolerable errors (%)	93.6	93.5	98.8	98.04
CMF and voting logic errors (%)	6.4	6.5	1.2	1.96

From the data in the table, we can highlight two interesting results. First, the percentage of faulty words per frame is higher in the RRR designs than in the RPR designs. This is possibly due to the fact that the RR modules are made with R2SER2SDF FFTs. This kind of architecture performs some calculations using a serial approach, which means that an error affecting one stage can accumulate in two successive calculations, whereas errors in the stage of a full-pipelined design only affect the results of that stage. Additionally, the calculations in the RRs could also be more prone to failure because they have full resolution. The second interesting result is the classification of erroneous words. We can see that a smaller fraction of the erroneous words corresponds to uncorrectable errors in the RRR architectures in contrast to the results of TMR and RPR architectures. The reduction in uncorrectable words is even relatively higher than the reduction in uncorrectable erroneous frames. Thus, the RRR technique produces a lower uncorrectable error rate with less uncorrectable errors per frame.

6.7. Conclusions

In this work we have proposed a novel error mitigation technique aimed at reducing the resources needed to mitigate errors. This technique is based on the addition of redundant modules that implement reduced resolution versions of the design to protect. The proposed RRR technique is general enough to be applied to a wide variety of algorithms. We demonstrated the application of this technique for an FFT design. For this case study, we demonstrated how to develop an optimal architecture, a proper conditioning of the

output signal and an adequate voting logic to detect and correct errors in the main module. Our experiments show that the error rate and the CMF rate are significantly improved with respect to the widespread Triple Modular Redundancy and other alternatives based on approximate calculations, such as Reduced Precision Redundancy. The Reduced Resolution Redundancy architectures are capable of significantly reducing the hardware overhead. They present a low sensitivity to errors, particularly to uncorrectable errors, and a good error correction capability, achieving a good level of noise in the voted output.

References

[176] M. Nicolaidis (Editor). “Soft Errors in Modern Electronic Systems”. Springer, 2010.

[177] K. A. LaBel and M. J. Sampson, “NEPP roadmaps, COTS, and small missions,” presented at the NEPP Electron. Technol. Workshop (ETW), Greenbelt, MD, USA, Goddard Space Flight Center, Jun. 2016.

[178] M. Nicolaidis, "Design for soft error mitigation," IEEE Transactions on Device and Materials Reliability, vol. 5, no. 3, pp. 405-418, Sept. 2005.

[160] M. Choudhury and K. Mohanram, "Low cost concurrent error masking using approximate logic circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 8, pp. 1163-1176, Aug. 2013.

[179] A. Sánchez-Clemente, L. Entrena, M. García-Valderas and C. López-Ongil, "Logic masking for SET Mitigation Using Approximate Logic Circuits," 18th International On-Line Testing Symposium (IOLTS), 2012, pp. 176-181, Jul. 2012.

[180] A. Sánchez-Clemente, L. Entrena and M. García-Valderas, "Error masking with approximate logic circuits using dynamic probability estimations," 20th International On-Line Testing Symposium (IOLTS), pp. 134-139, Jul. 2014.

[161] J. Sanchez-Clemente, L. Entrena, R. Hrbacek, L. Sekanina. "Error Mitigation using Approximate Logic Circuits: A Comparison of Probabilistic and Evolutionary Approaches". IEEE Transactions on Reliability, vol. 65, no. 4, pp. 1871-1883, Sep. 2016.

[97] B. Shim and N. R. Shanbhag, "Reduced Precision Redundancy for Low-power Digital Filtering," Proc. 35th Asilomar Conference on Signals, Systems and Computers, pp. 148-152, vol.1, 2001.

[98] B. Shim and N. R. Shanbhag, "Energy-Efficient Soft Error-Tolerant Digital Signal Processing," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 4, pp. 336-347, Apr. 2006.

[168] B. Pratt, M. Fuller and M. Wirthlin, "Reduced-Precision Redundancy on FPGAs," Int. Journal of Reconfigurable Computing, vol. 2011, Article ID 897189, 2011.

[105] S. Liu, K. Chen, P. Reviriego, W. Liu, A. Louri and F. Lombardi, "Reduced Precision Redundancy for Reliable Processing of Data," IEEE Transactions on Emerging Topics in Computing (in press), 2019.

[166] A. M. Keller and M. J. Wirthlin, "Partial TMR for Improving the Soft Error Reliability of SRAM-Based FPGA Designs", IEEE Transactions on Nuclear Science, vol. 68, no. 5, pp. 1023-1031, May 2021.

[1] L. A. García-Astudillo, A. Lindoso, L. Entrena, H. Martín, M. Garcia- Valderas. “Analyzing Reduced Precision Triple Modular Redundancy Under Proton Irradiation”. IEEE Nuclear & Space Radiation Effects (NSREC) Conf., Jul. 2021.

[157] E.H. Wold and A.M. Despain. “Pipeline and parallel-pipeline FFT processors for VLSI implementation”. IEEE Transactions on Computers Comput., C-33(5), pp. 414-426, May 1984.

[94] M. Berg, “Single Event Effects in FPGA Devices 2014-2015”, NASA Electronic Parts and Packaging Program (NEPP) Electronics Technology Workshop (ETW), June 2015.

[146] H. Quinn, K. Morgan, P. Graham, J. Krone, M. Caffrey, and K. Lundgreen, “Domain crossing errors: Limitations on single device triple-modular redundancy circuits in Xilinx FPGAs,” IEEE Transactions on Nuclear Science, vol. 54, no. 6, pp. 2037–2043, Dec. 2007.

[138] Xilinx Inc., “Zynq-7000 SoC Datasheet: Overview”, Datasheet DS190 (v1.11.1), Jul. 2018.

[74] Xilinx Inc., “Soft error mitigation controller v4.1,” Product guide PG036, Apr. 2018.

[2] L.A. García-Astudillo, A. Lindoso, L. Entrena, H. Martín, M. García-Valderas. "Error sensitivity study of FFT architectures implemented in FPGA", Microelectronics Reliability, pp. 114298, Oct. 2021.

7. EVALUATING REDUCED RESOLUTION REDUNDANCY FOR RADIATION HARDENING

The fault injection experiments carried out on the Reduced Resolution Redundancy technique showed its viability and hinted a very good performance with regards to the sensitivity of the hardened circuit to uncorrectable errors. For the experiments presented in this Chapter, we used two benchmarks: the one we used when the RRR technique was presented in Chapter 6 and a second, more complex, benchmark. This second benchmark was a mixed software and hardware implementation of an image processing algorithm proposed by ESA as a representative benchmark for space applications. The Reduced Resolution Redundancy mitigation technique was a very adequate choice for an image processing algorithm and its implementation proved to be a technical challenge as well as an interesting research path towards hardening MPSoCs.

With the idea of confirming the fault injection results we had obtained earlier, we performed neutron irradiation experiments on the FFT and the image processing benchmarks.

This chapter has been accepted for publication in IEEE Transactions on Nuclear Science after being presented at the RADECS 2022 Conference.

[5] © 2022 IEEE. Reprinted, with permission, from L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, *et al.*, "Evaluating Reduced Resolution Redundancy for radiation hardening," *IEEE Transactions on Nuclear Science*, 2023, (Early Access). DOI: 10.1109/TNS.2023.3268825

Abstract

Radiation Hardening By Design is traditionally performed using Triple Modular Redundancy, a very effective technique that introduces high overheads in terms of resources and power. The Reduced Resolution Redundancy technique presented in this work is a new Approximate Error Mitigation technique that uses redundant circuits with lower resolution to perform computations. In this work we evaluate this technique under radiation using two different benchmarks implemented in FPGA, namely a Fast Fourier Transform and an image processing algorithm for a near infrared detector. Experimental results from proton and neutron irradiation, and fault injection campaigns, show that the Reduced Resolution Redundancy hardening technique can effectively mitigate errors with reduced overheads.

7.1. Introduction

Radiation-induced soft errors are becoming an increasing concern for many applications, including not only those in space but also many at the ground level. Although there are effective solutions to tackle soft errors, many of them are not practical for less critical or low-cost applications. Radiation Hardening By Process (RHBP) is very expensive and not commonly available. Moreover, RHBP is only supported for rather mature technology nodes, because moving to smaller technology nodes requires a huge investment. As a result, it cannot take advantage of the higher performance and lower power consumption offered by the most advanced technologies used in commercial devices. Radiation Hardening by Design (RHBD) is an attractive alternative. However, it usually involves very large overheads. Partial redundancy solutions can be used to reduce the overheads, but finding an optimal trade-off with reliability is difficult in practice [175], [181], [166].

The increasing demand for performance is being solved by custom hardware architectures used to accelerate processing, which are often implemented in FPGAs. This is particularly relevant for Digital Signal Processing (DSP) applications. Again, choices of radiation-hardened FPGAs are very limited.

The new generations of FPGAs tend to be just radiation-tolerant, providing protection against the most critical radiation effects and requiring redundancy in the application logic to mitigate soft errors [182], [183]. Therefore, the redundancy that is needed to implement soft error mitigation prevents taking full advantage of the processing capabilities provided by these devices.

As a matter of fact, a fully redundant implementation (e.g., by Triple Modular Redundancy) may be overkill for many applications that have less stringent reliability requirements, or for ground applications, where the radiation levels are low. In these cases, we need to optimize the amount of redundancy to reach the required reliability levels without sacrificing too much of the power and performance budgets. Approximate redundancy solutions are being proposed for this purpose.

Approximate redundancy techniques try to strike an optimal balance between the extra redundant logic and the soft error mitigation. General solutions for random logic, such as those in [179], [180], [161], [163], [162], are based on selectively simplifying the redundant logic. The selection is assessed by error probabilities estimated along the simplification process [180]. The resulting approximate redundant circuit is not identical to the protected circuit, but it is able to mitigate most of the errors with reduced overhead.

DSP algorithms typically use complex data paths that require large amounts of resources, but we can take advantage of the nature of DSP computations to simplify them. Reduced Precision Redundancy (RPR) is a well-known approach that can produce good results by performing redundant computations with operands of reduced size [168]. The idea behind the Reduced Precision Redundancy technique is to add two copies of the design to protect, whose inputs have been truncated to a lower precision. This way the

Reduced Precision blocks need less resources and consume less power [97]. RPR designs on different granularity levels and digital circuits and components have been proposed in order to reduce power and resource consumption [184], [185]. The effectiveness and low noise of the RPR corrections have been tested in previous works under simulation experiments, demonstrating good theoretical results [100]. These results have recently been confirmed under radiation with protons [1], [3]. However, the application of reduced precision techniques may not be beneficial if the operators cannot be implemented with reduced precision.

Other hardening methods based on the premise of Reduced Precision Redundancy have been explored to obtain more precise results [3], [186], or lower resource consumption [106].

Reduced Resolution Redundancy (RRR) is an alternative Approximate Error Mitigation technique that has been proposed very recently [4]. This technique was demonstrated using fault injection, but it has never been tested under radiation. In this work we present the results of radiation experiments with low energy protons and compare them with the results obtained with RPR. The benchmark used for these experiments is a Fast Fourier Transform (FFT), which is an emblematic example of a DSP algorithm. A second irradiation campaign with neutrons was carried out to analyze the performance of this design under other type of particles.

To cover a more general case of RRR application, we also analyze a completely different benchmark proposed by the European Space Agency (ESA). This is a software benchmark that implements algorithms to process raw frames coming from a near infrared (NIR) HAWAII-2RG (H2RG) detector [187]. Using a hardware-software codesign approach, part of this benchmark was synthesized into hardware using a High-Level Synthesis (HLS) tool. Then, RRR was used to harden the hardware component. Finally, the system was implemented on a Zynq All Programmable SoC (APSoC), using the programmable logic (PL) for the hardened hardware component and the ARM-based processing system (PS) for the software component. Fault injection and neutron irradiation experiments were conducted to test this benchmark.

The rest of this paper is organized as follows. Section 7.2 introduces the Reduced Resolution Redundancy technique. Section 7.3 describes the experimental setup. Section 7.4 discusses the results of the experiments. Finally, Section 7.5 summarizes the conclusions of this work.

7.2. Reduced Resolution Redundancy

The Reduced Resolution Redundancy technique aims at hardening algorithms that work with sets of data of a particular size or resolution. To reduce the complexity in the redundant computations, RRR reduces the size of data sets, as opposed to RPR, which uses the same data sets but with reduced precision. By using redundant copies that

compute the same algorithm with a reduced subset of the input data, we can obtain an approximate result that can be compared with the Full Resolution (FR) data set. Thus, errors may be detected and corrected to some extent by using the Reduced Resolution (RR) results.

This technique is especially interesting when used in hardware implementations of DSP algorithms, because the reduction of resolution in the RR modules can be transformed into a reduction of the resources used to compute the algorithm and not just into a reduction of the time needed to compute the calculations, as it would happen in a software implementation. The reduction of the resources needed by the design benefits the power and resource consumption and makes the design less sensitive to errors, decreasing its cross section with respect to a Triple Modular Redundancy (TMR) implementation.

However, the RRR implementation requires a more complex data handling and conditioning of the RR results before they can be compared with the FR data sets. To reduce the resolution, the input data must be decimated. Decimation can be achieved either by ignoring some of the input data or by compressing the data in some way, such as averaging groups of data and using the averages as input for the RR module. The decimation performed in the input of the RR modules must be reversed at the outputs by interpolation. The easiest approach is to expand the results using Nearest Neighbor Interpolation (NNI), that is, replicating the value of the nearest data to fill the gaps in the expanded data set.

Figs. 7.1 and 7.2 show the block diagrams of two RRR implementations and how the data sets are treated before and after the calculations. Fig. 7.1 shows the RRR approach applied to an FFT design. In this case, the size of the RR input data is halved, and the resulting frame is expanded by repeating each data point of the frame until the Full Resolution size is achieved. This introduces some distortion in the conditioned RR result, but it can be compared with the FR result to detect and partially correct errors. Fig. 7.2 shows the RRR technique applied to an image processing algorithm. In this case, the RR input image is created by grouping neighbor pixels and performing the average of their values. The RR resulting image is expanded using NNI and used to detect errors in the FR image.

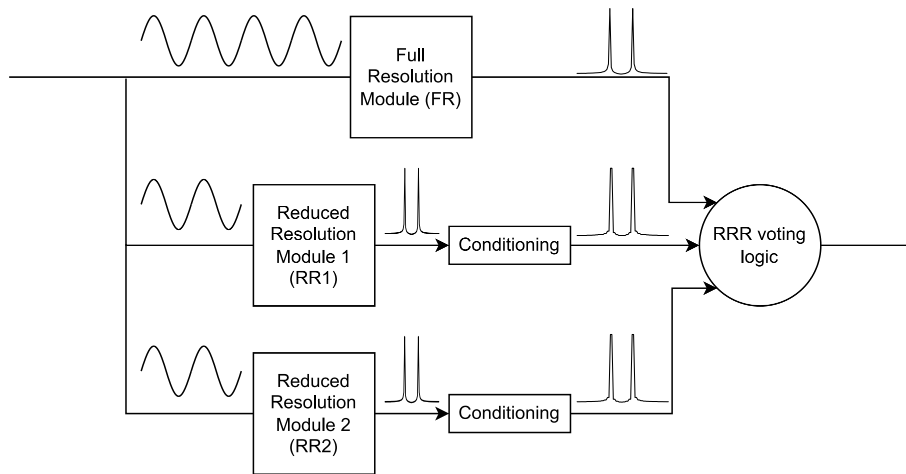


Figure 7.1: Block design of an FFT hardened with the RRR technique.

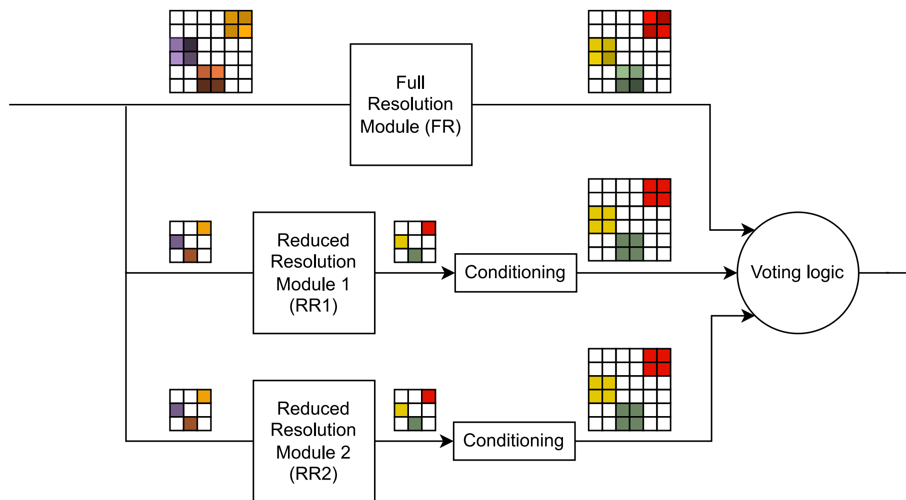


Figure 7.2: Block diagram of an image processing algorithm hardened with the RRR hardening.

The main objective of this approach is to reduce the resources used by other common hardening techniques. For hardware implementations, reducing the resolution usually implies performing less iterations, which would only reduce the time needed by the computation, not the resources used by the design. However, with some changes in the architecture of the Reduced Resolution modules, the higher throughput can be sacrificed in favor of the area overhead.

For regular DSP architectures, the application of RRR is straightforward, by simply reducing the amount of processing elements. In other cases, performing the necessary modifications may require knowledge of the operation of the circuit and may not be so easy. However, high-level synthesis tools can be used to optimize the architecture of the reduced resolution modules, as we show in this work.

Fig. 7.3 shows synthesis results for the two benchmarks that we tested in this work implemented in the FPGA fabric of a Zynq-7010 device . The graph presents the usage of Flip-Flops (FF), Look-Up Tables (LUT) and Digital Signal Processing (DSP) blocks

of the two FFT architectures and the two NIR HAWAII architectures that will be used as Full and Reduced Resolution modules. From the data in the graph, we can see that the resources are nearly halved in the RR versions of the selected benchmarks. Higher reductions of resources could be achieved with higher undersampling of the input data, but this would affect the quality of the RR results. Achieving a reduction of around half of the FR module is a well-balanced solution.

An advantage of RRR is that it can reduce the use of all types of resources, including those that have a fixed precision. For instance, the RR FFT design can be implemented with half of the DSP blocks of the FR design. In contrast, an RPR implementation cannot reduce the usage of DSP blocks, because the precision of the DSP blocks is fixed.

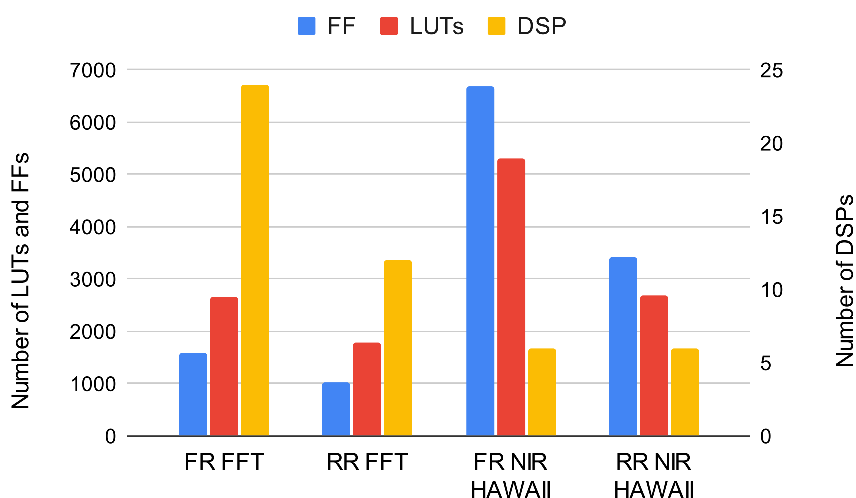


Figure 7.3: Utilization of hardware resources of Full and Reduced Resolution individual implementations of the FFT and NIR HAWAII algorithms.

The voting algorithm used to detect and correct errors leverages the same concepts used by an RPR voter. The differences between the FR and RR results are compared against a threshold. If an error greater than the threshold appears in the FR output, the final result is corrected with the RR result, provided both RR modules yield the same result. In any other case, the FR result is considered as correct and it is used as the final output. This approach produces a very good precision at the output, except for Common-Mode Failures (CMFs), that is, cases when more than one module is affected by a fault at the same time. Additional checks, other than the error thresholding, can be performed in the voter to ensure good quality results. In Fig. 7.4 we illustrate the basic RRR voter.

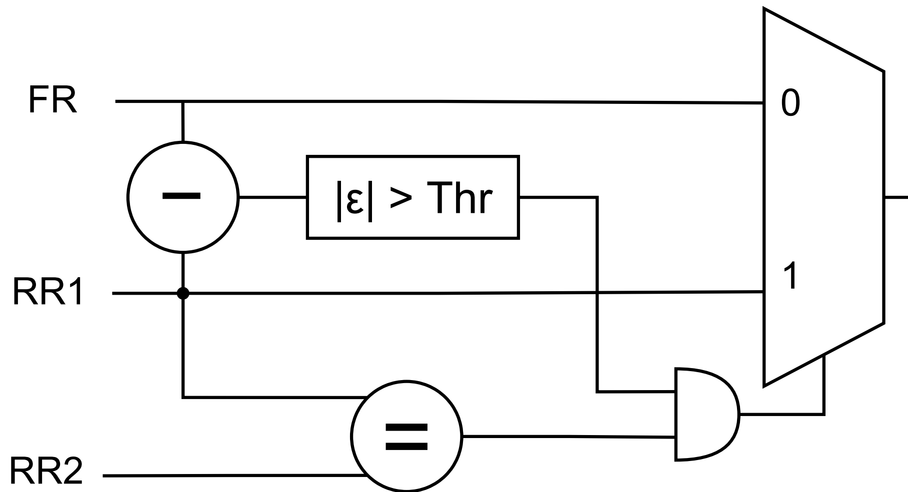


Figure 7.4: Diagram of a basic RRR voting logic.

7.3. Experimental setup

We implemented two Reduced Resolution Redundancy benchmarks based on the FFT and the NIR HAWAII designs.

The designs were implemented in a Xilinx Zynq-7010 All-Programmable System-On-Chip (APSoC) [138], which uses a 28 nm technology. The RRR FFT was implemented using just the programmable logic of the device. In the RRR NIR HAWAII benchmark, we used as well the ARM Cortex-A9 microprocessor in the processing system. Both designs were implemented using default tool options. In order to control the experiments, we used a Raspberry Pi board, that works as an external controller to collect the results of the experiment and perform a reset and reconfiguration of the FPGA as soon as an error is registered.

This same setup can be used for injection and radiation experiments with minor changes. We performed the injection campaigns using the Xilinx Soft Error Mitigation IP [74], which is able to perform injections in the configuration memory of the FPGA and correct the errors found by scrubbing the memory. Faults were injected at a sufficiently slow rate as to ensure they do not overlap. Only single bit upsets are considered.

The FFT benchmark was tested using 15 MeV protons at Centro Nacional de Aceleradores (CNA) in Seville and atmospheric neutrons at ChipIr (UK). The NIR HAWAII benchmark was only tested with neutrons at the same facility.

7.4. Experimental results

To test the Reduced Resolution Redundancy technique, we implemented two benchmarks hardening two different applications. This section covers the implementation details of the designs and the results of the irradiation and injection experiments performed with

these benchmarks.

7.4.A. FFT benchmark

The first benchmark we tested was an FFT implemented with a Radix-2 Single-path Delay Feedback (R2SDF) architecture. This is a pipelined architecture which is commonly used in high-performance hardware designs. It was hardened using two Reduced Resolution FFT modules, which use a specially tailored architecture called R2SER2SDF (Radix-2 Serial-2 SDF) [4]. This architecture computes a half-resolution FFT and increases the computation latency by serializing every second stage. This way, this module computes the RR algorithm in the same time used by the R2SDF FFT, but using less resources.

The basic Radix-2 FFT pipeline stage used in the Full Resolution module is depicted in Fig. 7.5.a, and the more complex R2SER2 modification is shown in Fig. 7.5.b. Each stage in the R2SDF FFT implementation (Fig. 7.5.a) consists of a butterfly unit, a FIFO buffer to delay data, and a complex multiplier. For a transform length N, the complete FFT design uses log2(N) stages. In the R2SER2SDF architecture, each stage (Fig. 7.5.b) also uses a butterfly unit and a complex multiplier, but the FIFO structure is more complex to serialize every two computations. Each R2SER2SDF stage is equivalent to two R2SDF stages. As a result, the complete R2SER2SDF FFT design uses only log2(N)/2 stages.

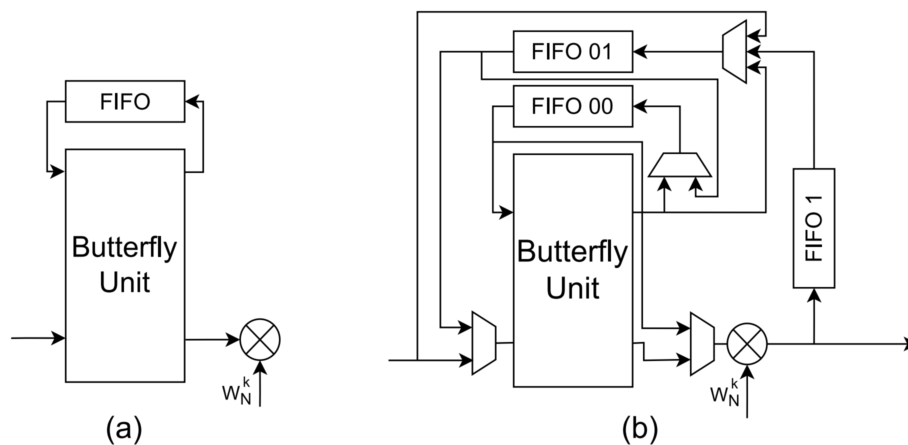


Figure 7.5: Pipeline stages of (a) a Radix-2 FFT implementation and (b) a Radix-2-Serial-2 FFT architecture.

Once the Reduced Resolution Redundancy system has been assembled, including the FR FFT and the two RR FFTs, the voting logic and other control logic, the final design still offers a significant reduction of resources. Fig. 7.6 shows the consumption of resources of the RRR FFT design in terms of LUTs, FFs and DSPs, compared to a TMR implementation, a Scaled Reduced Precision Redundancy implementation and a normal Reduced Precision Redundancy implementation. The Scaled RPR [3] is a special RPR that uses different precisions in each stage of the pipeline for a balance between accuracy and resource reduction. Instead of using 8-bit data throughout the FFT pipeline,

the Scaled RPR uses 8-bit data just in the first stage and the bit-width increases gradually until the whole 16-bit precision. As it can be seen in the figure, the usage of resources of the RRR FFT design is very similar to that in the Scaled RPR implementation, standing between the TMR and the complete RPR designs in terms of FFs and LUTs, but lowering the number of DSP blocks needed for all cases. The comparison of the RRR FFT with the Scaled RPR FFT will show us the advantages of the RRR technique, since these two are the most similar benchmarks in terms of resources.

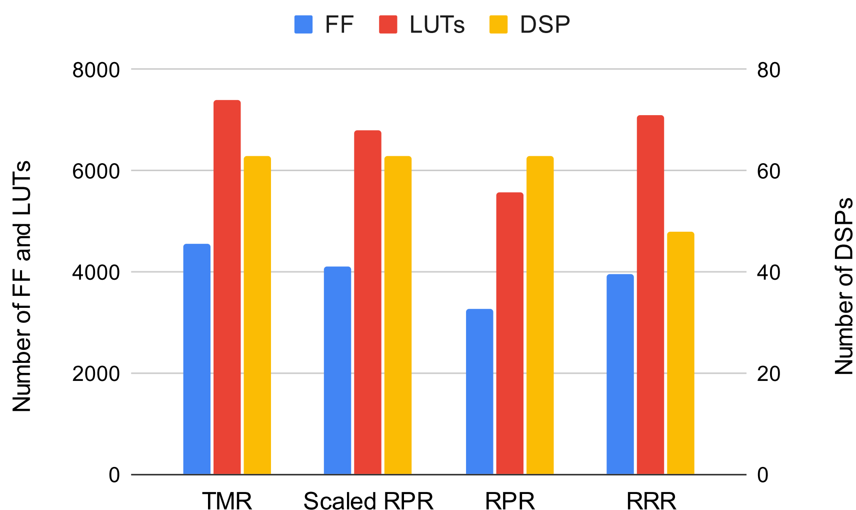


Figure 7.6: Resource consumption of different hardening techniques applied to the FFT benchmark.

In this RRR FFT implementation, the RR FFTs are fed with half of the input values and then the results are extended to the full resolution using Nearest Neighbor Interpolation by repeating each data point. The voting logic follows the schematic in Fig. 7.4, with the addition of a peak detection algorithm that ensures the most critical information of the FFTs (the peak frequency components) is not lost in the voting process.

Table 7.1 shows the results of the radiation experiments for a 256-point R2SDF FFT hardened using RRR. In the first column we present the results of proton irradiation and in the second column, the results under neutron irradiation. The results of an equivalent FFT using Reduced Precision Redundancy (RPR) [3] are added in the last column for the sake of comparison. The Table presents the number of erroneous FFT calculations (faulty frames), and how many of those could not be corrected by the voting logic. There are two types of uncorrectable errors: Common-Mode Failures, which affect two or more of the FFT modules at the same time, rendering the voting logic useless, and errors affecting the voting logic itself. The rest of the faulty frames contain errors that only affected the RR blocks and did not reach the voted output (masked errors), errors that were approximately corrected using the RR result (corrected errors) or errors in the FR block smaller than the threshold, which are considered tolerable errors.

We present the cross sections related to the total faulty frames and the uncorrectable

frames, respectively. These two results measure the overall sensitivity of the circuit and its sensitivity to errors that cannot be corrected by the mitigation technique, respectively. The overall cross-section is similar to the cross-section of an unprotected design, and it is typically two orders of magnitude higher than the uncorrectable cross-section. We show the 95% confidence intervals of the cross section between parentheses, calculated using a Gaussian distribution for the cross-section and a Chi-Square distribution for the uncorrectable cross-section, since the number of samples is lower. The next rows analyze the error distribution from the point of view of the faulty data points inside each FFT frame, classifying them in two categories: correctable data words are those in which the error was masked, corrected or classified as tolerable. On the other hand, uncorrectable data words are found when CMFs or voting logic errors affect a word in the frame.

The last row shows the Peak Signal-to-Noise Ratio (PSNR), a measure of the distortion present in the signal, calculated as in Eq. 7.1, where MAX denotes the higher possible value of a data point, and the Mean Squared Error (MSE) is the sum of the squares of the errors found in each frame. Higher values indicate less noise in the output.

Table 7.1: RADIATION RESULTS OF THE RRR FFT DESIGN AND COMPARISON WITH AN RPR FFT IMPLEMENTATION.

	RRR FFT (Triple voter) under proton irradiation	**RRR FFT (Triple voter) under neutron irradiation**	**Scaled RPR FFT (Triple voter) under proton irradiation**
Faulty frames	162	74	187
Uncorrectable frames	2	0	4
Cross-section (10^{-11} cm^2)	8.1 (6.9, 9.4)	92.7 (71.6, 114.3)	30 (25.8, 34.5)
Uncorrectable cross-section (10^{-12} cm^2)	1.0 (0.1, 3.6)	0 (0 ,3.7)	9.6 (3.6, 21.1)
Total words in faulty frames	82,944	37,888	95,744
Faulty words	17,038 (20.5%)	6,934 (18.3%)	17,145 (17.9%)
Correctable (%)	99,5	100	94
Uncorrectable words (%)	0,5	0	6
PSNR (dB)	71.8	68.7	90.6

$$PSNR = 10 \times log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{7.1}$$

By analyzing the data from the proton irradiation experiments, we can see that they yield very similar results to the injection campaigns reported in [4]. When comparing these results with the results of a proton irradiation experiment in a similar Reduced Precision Redundancy, we can see that the RRR benchmark performs better in both cross section and uncorrectable cross section. The uncorrectable error cross section is especially low in comparison to Scaled RPR, almost ten times smaller. The Scaled RPR has been shown to have to have a cross-section close to TMR [3]. This result confirms the results previously obtained by fault injection [4], which show an uncorrectable error sensitivity 3.6 times lower than the TMR design. In terms of the error classification, we observe that the error correction capabilities of RRR for this benchmark are remarkable, reaching a correction rate of more than 99% of the faulty words. However, the results corrected by

interpolated results take their toll on the noise of the voted output, when comparing the PSNR to the RPR correction.

The neutron irradiation experiments shown in the second column reproduce a similar behavior to that of the proton experiments. Although the number of samples is not as high as in the proton case, the test showed no uncorrectable frames, which hints at a very low sensitivity to uncorrectable errors. The error classification is quite comparable in both cases and the PSNR, although a bit lower in the neutron experiment, is approximately the same, assuming some uncertainty level.

Fig. 7.7 shows more detailed information about the error classification of the words found in the faulty frames during the radiation campaigns. The strikingly low amount of uncorrectable errors in the RRR design is very apparent in the figure. In this graph we can also see the reason behind the slight difference in the PSNR between the RRR under proton and under neutron irradiation: in the neutron experiment, the amount of completely corrected (masked) errors is lower, thus, the final output is noisier than in the proton experiment. However, the higher PSNR in the RPR design is explained by the more precise data used to correct, not by the percentage of completely masked errors.

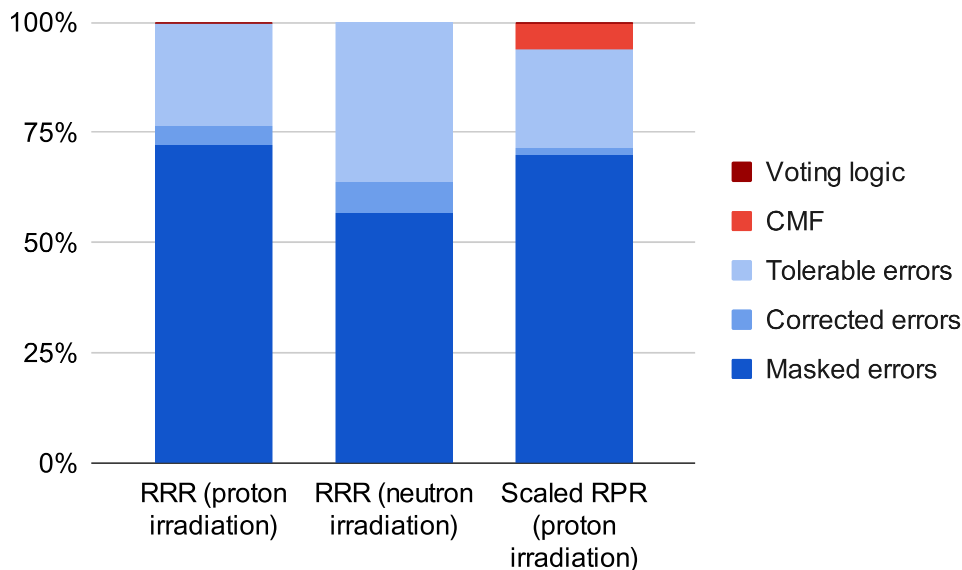


Figure 7.7: Error classification of the faulty words in the FFT benchmarks.

7.4.B. NIR HAWAII image processing benchmark

The near infrared (NIR) HAWAII-2RG (H2RG) benchmark is an image processing algorithm aimed at the detection of galactic cosmic rays by the analysis of certain parameters in groups of images taken by a NIR sensor [187]. This benchmark has been proposed by ESA as a significant example for profiling and evaluating performance of alternative implementations. The algorithm involves various steps in which the initial image suffers some transformations and a final computationally heavy step called

detectCosmicRay, where some image characteristics are extracted. The process is repeated a number of times with different images and a statistical analysis is performed in order to determine the presence of a galactic cosmic ray. A software implementation of the NIR HAWAII benchmark has been recently tested under fault injection experiments using soft error mitigation techniques to protect the microprocessor against SEUs [144].

In our experiments, we used 512×512-pixel synthetic images, which were randomly generated within the benchmark with a function provided by the developers.

Our implementation involves the processing system and the programmable logic of the Zynq device. The input image generation and the initial steps of the algorithm are calculated in the microprocessor, whereas the computationally heavy detectCosmicRay function is accelerated in hardware and its results are sent back for the microprocessor to compute the final statistical analysis.

The image transfer from the microprocessor to the programmable logic for acceleration and back to the microprocessor was made through the shared memory of the Zynq. A Direct Memory Address (DMA) engine was in charge of the data transfer.

To implement the RRR system, we synthesized two versions of the detectCosmicRay method from C language to a hardware IP using Xilinx Vitis High-Level Synthesis (HLS) tool. The first version, intended for the Reduced Resolution modules, is constrained to use a single divider hardware block, while the second version, the one intended for the Full Resolution module, is constrained to use two dividers. With this approach, the three modules need approximately the same number of clock cycles to complete the calculations, but the FR module needs about twice as many resources. Table 7.2 summarizes the HLS report of the FR and RR synthesized blocks. We can observe in the table that the execution clock cycles estimation, given by the iteration latency, is similar for both designs, while the flip-flops and LUTs counts are nearly halved in the Reduced Resolution implementation.

Table 7.2: HIGH-LEVEL SYNTHESIS TIMING AND RESOURCES ESTIMATES FOR THE DETECTCOSMICRAY FUNCTION

	Clock period estimate (ns)	Iteration latency	Iteration interval	DSP	FF	LUT
detectCosmicRay_FR	8.51	136	3	6	9,119	7,153
detectCosmicRay_RR	8.51	149	5	6	4,267	3,554

The results of the three detectCosmicRay blocks are sent to the microprocessor memory using DMA. Then, the interpolation of the RR results and the voting of the pixels are implemented in the microprocessor by software means. The interpolation is performed using the NNI technique, just by expanding the value of each RR pixel in its three neighbor pixels (the right pixel, the bottom pixel and the pixel in the right-down diagonal). Once interpolated, the resultant images and the FR image are compared, and their values are voted pixel-to-pixel with the voting approach previously explained and shown in Fig. 7.4.

For error detection and analysis purposes, the detectCosmicRay function is also computed in software and its results are compared with the voted results of the data coming from the programmable logic. In case of discrepancies, an error is reported and sent to the host computer for further analysis.

Because of the high consumption of resources, we could not implement the RRR technique to harden the whole NIR HAWAII algorithm. Instead, we just hardened the detectCosmicRay function, as explained previously. For the same reason, a TMR solution can be designed but cannot be implemented in the Zynq device we used, because even triplicating just the detectCosmicRay method was too expensive. Fig. 7.8 shows the utilization of resources of the RRR benchmark previously described and a comparison with the TMR design of the same system. As is apparent from the figure, the number of flip-flops and LUTs is reduced significantly using the RRR technique, although the number of DSP blocks used is the same. These results follow the estimations made by the HLS software, but the optimizations performed by the tool in the final implementation have reduced the final number of flip-flops and LUTs.

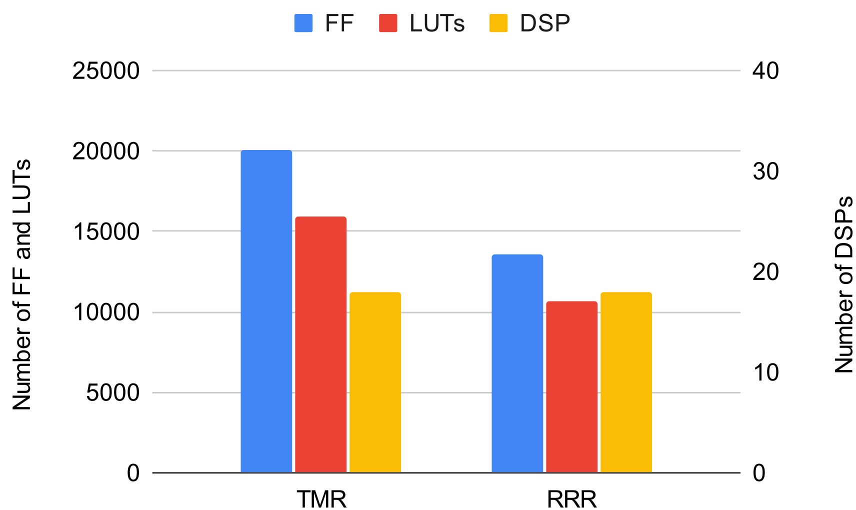


Figure 7.8: Post synthesis resource consumption of the TMR and RRR hardening techniques applied to the NIR HAWAII benchmark.

The results presented in the Table 7.3 show the performance of the RRR error mitigation technique in comparison with fault injection results from the FFT RRR benchmark presented previously. The fault injection results are presented here in a very similar way as in the radiation experiments, with the exception of the metrics we call Error rate and Uncorrectable error rate, which are devised as measurements of the sensitivity of the design, in a similar fashion that the cross section and uncorrectable cross section measure the sensitivity in radiation experiments. The error rates are calculated as the number of faulty frames or uncorrectable faulty frames divided by the number of addresses injected in the configuration memory.

With regard to the injections, it is worth noting that errors injected in the configuration

memory only affect the calculations performed in programmable logic, including the DMA engine responsible for the data transfers to and from the processing system. We also configured the SEM IP to ensure a single error is injected per image processed in the algorithm, avoiding multiple fault injection.

The data in the table shows promising behavior in terms of error sensitivity and error correction capabilities. When comparing these results with those of the RRR FFT injection experiments, which we presented in [4], we can see that the error classification of the pixels or words contained in the faulty calculations of the algorithms is quite similar. Other metrics, such as the error rates or the PSNR are not comparable across the algorithms, because of the different utilization and distribution of the resources or the very nature of the algorithm.

Table 7.3: INJECTION RESULTS OF THE RRR NIR HAWAII DESIGN AND COMPARISON WITH AN RRR FFT IMPLEMENTATION.

	RRR NIR HAWAII	**RRR FFT**
Number of injections	294,000	360,000
Faulty images	758	1,352
Uncorrectable images	22	14
Error rate ($\times 10^{-3}$)	2.58	3.76
Uncorrectable Error rate ($\times 10^{-5}$)	7.48	3.92
Total pixels in faulty images	198,705,152	692,224
Faulty pixels	70,257,520 (35.4%)	155,939 (22.5%)
Correctable (%)	96	98.8
Uncorrectable pixels (%)	4	1.2
PSNR (dB)	140	65.8

Table 7.4 shows the results of the irradiation campaign of the RRR NIR HAWAII design under a neutron beam. The results are presented in this table in the same way we presented the radiation results for the RRR FFT experiments.

Table 7.4: RADIATION RESULTS OF THE RRR NIR HAWAII DESIGN UNDER NEUTRON IRRADIATION AND COMPARISON WITH THE RRR FFT IMPLEMENTATION

	RRR NIR HAWAII	**RRR FFT**
Faulty images	212	74
Uncorrectable images	51	0
Cross section (10^{-10} cm^2)	6.8 (5.9, 7.7)	9.27 (7.2, 11.4)
Uncorrectable Cross section (10^{-10} cm^2)	1.6 (1.3, 4.3)	0 (0, 3.7)
Total pixels in faulty images	55,574,528	37,888
Faulty pixels	29,264,340 (52.7%)	6,934 (18.3%)
Correctable (%)	74.5	100
Uncorrectable pixels (%)	26.5	0
PSNR (dB)	122.1	68.7

As mentioned earlier, the fault injection method did not allow us to inject faults in the microprocessor. Thus, voting logic errors do not appear in the injection experiments. However, in the irradiation experiments, particles affected the processing system as well as the programmable logic. For this reason, the radiation experimental results do not match the injection results very closely. The voting errors and other errors caused in the unhardened microprocessor may significantly affect the error distribution and the sensitivity of the circuit with respect to the injection results.

We performed a thorough quantitative analysis of the raw data in order to discriminate faults that can be clearly attributed to errors in the microprocessor part of the device, before analyzing the performance of the RRR benchmark in the FPGA. In particular, we could identify this kind of errors because of the extremely low amount of faulty pixels in the image, which was very different from the behavior exhibited in the fault injection error patterns. Nevertheless, even after the pre-processing of the data logs, we can observe in the table a notable increase in the number of uncorrectable calculations of the NIR algorithm in this radiation campaign. Around 25% of the faulty calculations were due to voting logic errors and CMF, a behavior that does not correspond to the results of the injection campaign and can also be easily attributed to faults in the microprocessor. The absence of error detection and correction mechanisms in the part of the algorithm executed in the microprocessor is responsible not only for the increase in the uncorrectable cross section of the design, but also of the deterioration of the quality of the voted signal, as we can see by the decrease in the PSNR of this experiment. An analysis of the total amount of errors attributed to the microprocessor (those with low amount of faulty pixels and the uncorrectable errors) gives a cross section of $7.5 \times 10^{-9} cm^2$, which is similar to the cross section reported in [188] for the same unhardened processor running other complex benchmarks. To confirm this hypothesis, we performed a complementary experiment. We implemented a version of the NIR HAWAII circuit, eliminating all the tasks performed by the microprocessor and the hardware resources devoted to communication. The input

generation and output checking of the RRR modules were implemented in the FPGA in this version and a basic warning system was implemented to flag errors in any resultant image. We irradiated this design under the same neutron beam at ChipIr and we found out that, of all the 2485 faulty calculations detected, only one of them was a CMF (cross section of $8.41 \times 10^{-13} cm^2$), the rest of them being simple faults in one of the modules. This experiment confirms that the vast majority of the uncorrectable faults, and maybe some of those that could be corrected, in the complete NIR HAWAII RRR, were caused by the unprotected microprocessor and the communication hardware.

In the second column of Table 7.4, we show again the results of the neutron irradiation of the RRR FFT benchmark for comparison purposes. We observe an important difference in the cross section of both designs, but this is a matter of the benchmark tested and how it is implemented. This comparison does not add any extra information, except confirming that the presence of faults in the microprocessor must be prevented before attempting further irradiation experiments in this benchmark.

As a final remark, we present in Fig. 7.9 the error classification of the two RRR benchmarks, comparing the injection and irradiation results. We have also added the error classification of the complementary experiment we performed using only the FPGA part (RRR NIR HAWAII FPGA only). Note that in this case we did not differentiate between tolerable, corrected or masked errors. We see a strong correlation in the RRR FFT experiments and, as seen previously, a significant difference in the NIR HAWAII experiments due to lack of protection in the microprocessor and the DMA channels. By extrapolating the behavior of the FFT experiments to the NIR HAWAII results, we could expect similar error percentages in the neutron irradiation results as in the fault injection experiments, provided we were able to mitigate the faults caused in the microprocessor hardware of the Zynq device. The error classification of the FPGA-only experiment confirms this assumption.

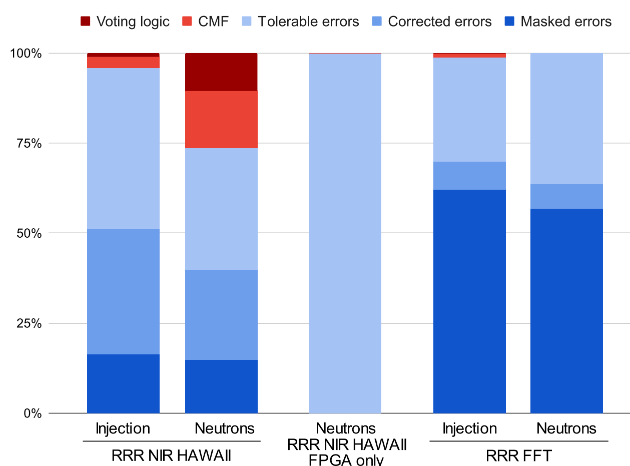


Figure 7.9: Error classification comparison between the fault injection and radiation campaigns in the RRR designs.

Assuming that the majority of uncorrectable errors in the NIR HAWAII experiments were due to the DMA communication (in the fault injection campaign), and a combination of errors in the communication and in the microprocessor (in the neutron irradiation campaign), we then compared the error classification results after removing the uncorrectable errors, as seen in Fig. 7.10. The error classification matches almost perfectly for the two experiments. This reinforces the necessity of hardening both the microprocessor and the interface between the processing and the programmable logic areas of this kind of devices.

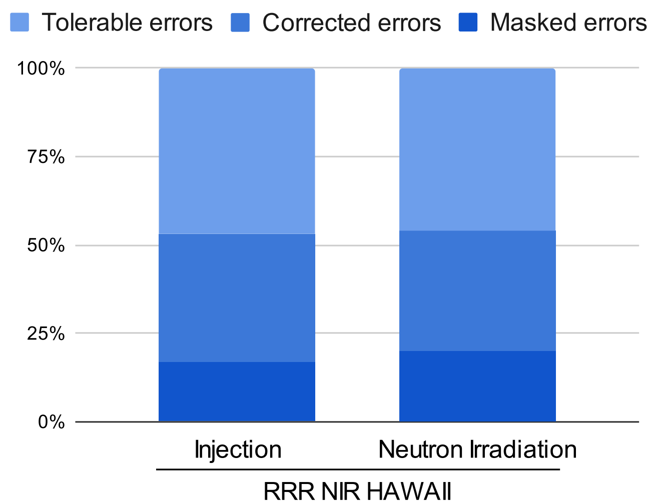


Figure 7.10: Error classification of the NIR HAWAII results in fault injection and neutron irradiation after removing the uncorrectable errors found.

7.5. Conclusions

In this work we performed irradiation and injection experiments to evaluate the error mitigation capabilities of the Reduced Resolution Redundancy technique, which is a novel Approximate Error Mitigation technique that was recently proposed. This technique aims at reducing the amount of redundant resources that are required by traditional error mitigation techniques, such as Triple Modular Redundancy.

The RRR technique relies on the addition of redundant modules that compute an algorithm over decimated data sets and the usage of interpolations to permit the comparison and voting. We can trade off the execution time gained by processing reduced sets of data for a lower consumption of resources. This technique is very well suited for digital signal processing algorithms which can use a selectable resolution.

Based on this approach, we conducted experiments on two benchmarks: a hardware-based Fast Fourier Transform and an image processing algorithm in a hybrid hardware and software implementation. Our results show that this approach may be used as an alternative to the widespread TMR, since it is able to significantly reduce the area

overhead and offers good error correction capabilities without compromising the precision of the results. The RRR technique also achieves a better error correction rate than other approximate techniques, such as RPR. The experiments show that the cross section for protons and neutrons is significantly reduced with respect to RPR, even in the case the amount of used resources is similar.

7.6. Acknowledgments

The authors want to thank ChipIr team for their help during the experiments.

References

[182] T. Jones. “Xilinx Technologies for New Space/Space 2.0”. Technical Education Webinar Series, Oct. 2021. Available at: https://www.microwavejournal.com/events/2132-xilinx-technologies-for-new-space-space-2-0

[183] “Radiation-Tolerant FPGAs. Space Solutions”, Microchip brochure DS00003023C, 2020.

[179] A. Sánchez-Clemente, L. Entrena, M. García-Valderas and C. López-Ongil, "Logic masking for SET Mitigation Using Approximate Logic Circuits," 18th Int. On-Line Testing Symp. (IOLTS), pp. 176-181, Jul. 2012.

[180] A. Sánchez-Clemente, L. Entrena and M. García-Valderas, "Error masking with approximate logic circuits using dynamic probability estimations," 20th Int. On-Line Testing Symp. (IOLTS), pp. 134-139, Jul. 2014.

[161] J. Sanchez-Clemente, L. Entrena, R. Hrbacek, L. Sekanina. "Error Mitigation using Approximate Logic Circuits: A Comparison of Probabilistic and Evolutionary Approaches". IEEE Trans. on Reliability, vol. 65, no. 4, pp. 1871-1883, Sep. 2016.

[163] A. J. Sanchez-Clemente, L. Entrena, and M. Garcia-Valderas, “Partial TMR in FPGAs using approximate logic circuits”, IEEE Trans. on Nuclear Science, vol. 63, no. 4, pp. 2233–2240, Aug 2016.

[162] A. J. Sánchez Clemente, L. Entrena y F. Kastensmidt, "Approximate TMR for selective error mitigation in FPGAs based on testability analysis," 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2018.

[168] B. Pratt, M. Fuller and M. Wirthlin, "Reduced-Precision Redundancy on FPGAs," Int. Journal of Reconfigurable Computing, vol. 2011, Article ID 897189, 2011.

[97] B. Shim and N. R. Shanbhag, "Reduced Precision Redundancy for Low-power Digital Filtering," Proc. 35th Asilomar Conference on Signals, Systems and Computers, pp. 148-152 vol.1, Nov. 2001.

[184] P. Reviriego, J.A. Maestro, I. López, J. A. de Agapito. “Soft error tolerant infinite

impulse response filters using reduced precision replicas". Proc. European Conf. on Radiation and Its Effects on Components and Systems, RADECS, 493–496, March 2012.

[185] K. Chen, L. Chen, P. Reviriego, F. Lombardi, "Efficient implementations of reduced precision redundancy (RPR) Multiply and accumulate (MAC)". IEEE Trans. on Computers, 68(5), pp. 784–790, Dec. 2018.

[100] M. A. Sullivan, H. H. Loomis, A. A. Ross, "Employment of reduced precision redundancy for fault tolerant FPGA applications", Proc. IEEE Symp. on Field Prog. Custom Comp. Machines (FCCM), pp. 283–286, Oct. 2009.

[1] L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, H. Martín, P. Martin-Holgado, M. Garcia-Valderas, "Analyzing Reduced Precision Triple Modular Redundancy Under Proton Irradiation," IEEE Trans. on Nuclear Science, vol. 69, no. 3, pp. 470-477, March 2022.

[3] L. A. García-Astudillo, A. Lindoso, L. Entrena, H. Martín, M. García-Valderas and P. Martín-Holgado, "Analyzing Scaled Reduced Precision Redundancy for Error Mitigation under Proton Irradiation," IEEE Trans. on Nuclear Science, vol. 69, no. 7, pp. 1485-1491, July 2022.

[186] W. Stechele "Protecting FPGA-based automotive systems against soft errors through reduced precision redundancy", Proc. 10th IEEE Int. Symp. on Ind. Embedded Syst. (SIES), pp. 170–173, Aug. 2015.

[106] F. F. dos Santos, M. Brandalero, M. B. Sullivan, P. M. Basso, M. Hubner, L. Carro, P. Rech, "Reduced Precision DWC: An Efficient Hardening Strategy for Mixed-Precision Architectures", IEEE Trans. on Computers, 71 (3), 573–586, Feb. 2021.

[4] L. A. García-Astudillo, L. Entrena, A. Lindoso, H. Martín, "Reduced Resolution Redundancy: A Novel Approximate Error Mitigation Technique," IEEE Access, vol. 10, pp. 20643-20651, Feb. 2022.

[187] A. Jung, P.-E. Crouzet, "The H2RG infrared detector: introduction and results of data processing on different platforms", ESA. Available at: https://essr.esa.int/project/nir-hawaii-2rg-data-processing-algorithms-benchmarking-software

[138] Xilinx Inc., "Zynq-7000 SoC Datasheet: Overview", Datasheet DS190 (v1.11.1), Jul. 2018.

[74] Xilinx Inc., "Soft error mitigation controller v4.1," Product guide PG036, Apr. 2018.

[144] P. M. Aviles, L. Schäfer, A. Lindoso, J. A. Belloch, L. Entrena, "High complexity reliable space applications in commercial microprocessors", Microelectronics Reliability, 114679, September 2022.

[188] A. Lindoso, M. Garcia-Valderas, L. Entrena, "Analysis of neutron sensitivity and data-flow error detection in ARM microprocessors using NEON SIMD extensions", Microelectronics Reliability, 100–101, 113346, Sept. 2019.

[175] B. Pratt, M. Caffrey, J. F. Carroll, P. Graham, K. Morgan, M. Wirthlin, “Fine-grain SEU mitigation for FPGAs using partial TMR”, IEEE Trans. on Nuclear Science,vol. 55, no. 4, pp. 2274–2280, Aug. 2008.

[181] O. Ruano, J. A. Maestro, P. Reviriego, “A methodology for automatic insertion of selective TMR in digital circuits affected by SEUs”,IEEE Trans. on Nuclear Science,vol. 56, no. 4, pp. 2091–2102, Aug. 2009.

[166] A. M. Keller, M. J. Wirthlin, "Partial TMR for Improving the Soft Error Reliability of SRAM-Based FPGA Designs," IEEE Trans. on Nuclear Science, vol. 68, no. 5, pp. 1023-1031, May 2021.

8. OPTIMIZED REDUNDANCY FOR COMPOSITE ALGORITHMS

8.1. Introduction

Further research on the topic of Approximate Error Mitigation led us to the development of the Optimized Redundancy for Composite Algorithms (ORCA), the last original contribution of this Thesis. This novel hardening technique has been conceived to mitigate errors in hardware implementation of algorithms that may be decomposed into more simple parts. Circuits hardened using this technique are able to perform exact corrections under certain circumstances, instead of approximate, which is a remarkable achievement, since the corrections introduce less distortion in the result compared to other Approximate Error Mitigation techniques. Instead of adding identical redundant modules to compare the results and detect possible mismatches, for this technique we use complementary modules that can be composed to implement the target design. These complementary modules can also be compared to detect errors. When they are correct, they produce an exact result instead of an approximate result.

We have selected two representative digital signal processing benchmarks to test the application of the ORCA technique in an FPGA: the Fast Fourier Transform (FFT) and a Finite Impulse Response (FIR) filter. The experimental results show that this technique can reduce the overhead compared to TMR implementations and provide a better trade-off between overhead and precision than existing approximate techniques. The proposed approach is very general and can be applied to a wide variety of composite algorithms.

The findings presented in this Chapter are presently under review for publication in IEEE Transactions on Aerospace and Electronic Systems.

The rest of this Chapter is organized as follows. Section 8.2 presents related background on Approximate Error Mitigation and the motivation of this work. Section 8.3 describes the Optimized Redundancy for Composite Algorithms. In Section 8.4 we discuss the implementation of the technique in two case studies: the FFT and the FIR filter. The experimental setup is described in Section 8.5. In Section 8.6 we present the results from the fault injection experiments we performed. Finally, Section 8.7 shows the conclusions of this work.

8.2. Background and related work

Among the trends of the so-called New Space paradigm is the usage of Commercial Off-The-Shelf parts instead of rad-hard devices. As long as an acceptable reliability can be achieved, the advantages of COTS in complexity and computing power make them very

attractive for low-cost missions. When hardened using RHBD techniques, COTS may be good candidates for applications where a strict dependability can be exchanged for higher performance or for economic reasons. COTS FPGAs are especially appealing for the implementation of custom aerospace applications because of their concurrent computing capabilities and flexibility.

As we have reviewed in previous Chapters, the basic error correction mechanism for digital circuits is Triple Modular Redundancy. TMR introduces two redundant copies of the target component and performs a majority voting to mask a single error in any of the copies. TMR can be applied at different hierarchy levels and granularities, ranging from the triplication of critical elements, such as registers (Local TMR), to the triplication of all combinatorial and sequential elements and majority voters (Distributed TMR). Triplication of whole blocks and voting just the outputs of the replicated modules (Block TMR) is often used to reduce the voting overheads. Adding more than two copies (N-Modular Redundancy) allows for correction of multiple simultaneous errors at the cost of higher overheads. These techniques provide a good error correction performance, but also create significant overheads in the area and power needed by the design. The development of Approximate Error Mitigation Techniques has been proven to be a way to reduce the overheads in previous Chapters of this Thesis. Approximate TMR proposes using approximate redundant copies of the target design, which can be implemented with less resources, at the expense of slight reductions in the precision of the result in case of error. Alternatively, other solutions are focused on selectively applying TMR to only a subset of components [166], [175], leaving the design unprotected for some errors.

The Reduced Precision Redundancy technique, also reviewed in Chapters 4 and 5, is a well-known approach in which the redundant blocks perform computations with less precision data [98], [168], [103], [170]. The results of these Reduced Precision modules are, thus, approximate versions of the Full Precision module. The voting of RPR systems ensures that any error in the FP module is corrected using a less precise, yet correct, result.

The RPR technique has been proven to be less sensitive than TMR to the effects of radiation while achieving good error correction capabilities, and a small incidence of uncorrectable errors, meaning Common-Mode Failures (CMF) and voting logic errors [1]. As we saw in Chapter 5, performing a non-uniform scaling of the data throughout the pipeline in an RPR-hardened system allows for better precision in the corrected results, with only a minor increase of the resources with respect to unscaled RPR [3].

Reduced Resolution Redundancy, the Approximate Hardening technique presented in Chapters 6 and 7, is based on the addition of two reduced resolution redundant blocks, which use a decimated set of input samples to perform the same calculation as the target module. Using an upsampling algorithm, the results of the Reduced Resolution modules can be compared with the Full Resolution result, correcting errors with approximate values when necessary. The voting logic is the same as the one used for RPR, but it may be complemented with additional checks to cover algorithm-dependent

cases, as the frequency peak detection logic proposed in Chapter 6. The injection and irradiation experiments performed in this Thesis in various benchmarks hardened using RRR demonstrate that this technique has an outstandingly low error sensitivity and low noise in the output of the approximately corrected computations and substantially reduces the area overhead of the hardened circuit [5].

The redundant modules added in all of these redundancy techniques must be identical. In all the previous Approximate Error Mitigation techniques, the system first verifies that the outputs of the redundant modules are the same. This condition is required to make sure that the approximate value used to check the full precision result is correct. An alternative approach proposed in [95] applies the concept of Design Diversity Redundancy (DDR) in a TMR scheme. However, the modules used in this approach must be functionally identical, although they differ in the implementation.Thus, they must provide the same output and cannot yield resource savings.

The ORCA technique we present in this Chapter utilizes two redundant modules different to each other, each of them also computing a different subset of the input data and whose results can be composed to create an exact result. This way, we can obtain both resource reduction and better quality corrections.

8.3. Optimized Redundancy for Composite Algorithms (ORCA)

In this section we describe the proposed Optimized Redundancy for Composite Algorithms (ORCA) error mitigation approach. As in the case of other Approximate Error Mitigation techniques, such as RPR or RRR, the ORCA technique is based on the addition of two redundant modules to the design to protect. In order to reduce the amount of resources needed, these additional modules do not calculate exact results, but approximations that can be compared to the exact result of the target design to detect and correct errors.

The ORCA error mitigation technique is conceived as a way to harden algorithms that can be computed using a composite or divide-and-conquer approach. This kind of algorithms can be decomposed into simpler parts that may be implemented separately and whose outputs are merged to obtain the final result. Typical examples of divide-and-conquer algorithms include large multiplication algorithms, image processing or the Fast Fourier Transform.

The main condition for the application of ORCA is that the algorithm allows decomposition into bounded components that can be computed in parallel, leveraging the concurrent computing capabilities of hardware implementations. In contrast to other approaches, ORCA does not use two identical redundant instances of an approximate design, but two different modules that can be combined to obtain the full result.

Fig. 8.1 illustrates the basic ORCA system. It contains a full module F and two reduced modules, A and B. Contrary to the RPR case, the two reduced modules are

complementary, not identical. A full implementation G can be obtained by composition of the A and B results. A and B must satisfy some kind of constraint that is used to detect errors. The usual constraint is that the results of A and B are similar, but not necessarily identical. A thresholded comparison may be used to check this kind of constraint.

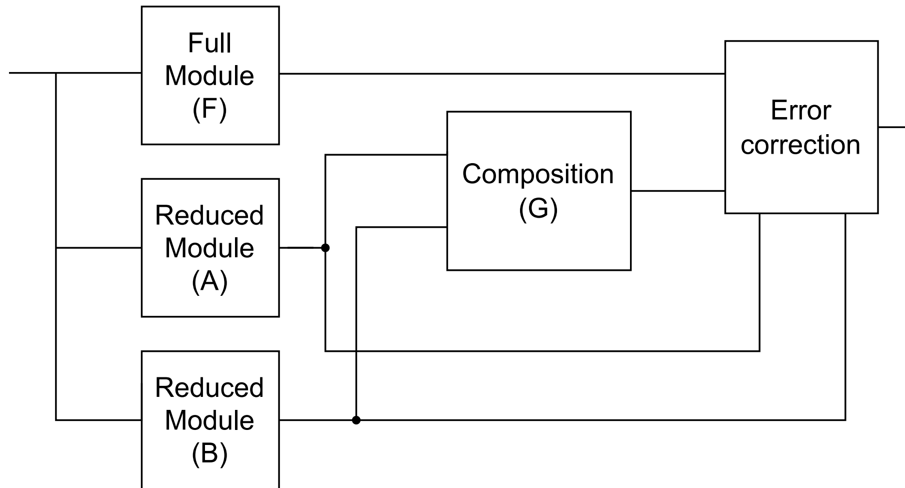


Figure 8.1: Block diagram of a basic Optimized Redundancy for Composite Algorithms design.

Error correction can be implemented with the same approach used for the RPR hardening technique, which was extensively discussed in previous Chapters: if the reduced modules A and B are similar, and F is not equal to G, then we take G as the correct output. Otherwise, we take the output from F. Alternatively, the comparison between F and G can be eliminated, provided that the composition operation can be hardened in some way (by triplication, for instance). Under this premise, the error correction algorithm works as follows:

1. If the outputs of A and B satisfy the similarity constraint, they are considered correct and the result is obtained from the composite result G.
2. If A and B do not satisfy the constraint, then the result is obtained from the full module F.

Comparing the ORCA approach with the previously presented Approximate Error Mitigation techniques from a theoretical point of view is interesting. The existing approximate approaches only have a full module (F) and always produce an approximate result when this module has an error. When the error in the F module exceeds the threshold, then the approximate result from the other modules must be taken. If the error is below the threshold, then the F result is taken, which is also approximate because it contains some error. Only when F has no error the result can be exact. To the contrary, with ORCA we compute two exact results, and it is still possible to obtain an exact result when any of them has an error. If the F result has an error, we take the output from G, which is exact. If any of the reduced modules, A or B, has an error that violates the

constraint, the correct result from F is taken. An approximate result is obtained only when the error produces a small variation in the output of the reduced modules, A or B, so that the threshold constraint is satisfied.

8.4. The FFT and FIR filter case studies

To evaluate the ORCA hardening technique, we created FPGA implementations of two typical algorithms that are widely employed in Digital Signal Processing: a Fast Fourier Transform and a digital FIR filter. They were implemented for high-performance, using a pipelined architecture. These algorithms are good candidates for the technique, not only for their ubiquity, but also because they may be implemented in such a way that they can be decomposed into simpler modules whose outputs are combined to create the full output.

In the following sections we describe the implementation of the algorithms and how to adapt the ORCA technique to harden them.

8.4.A. FFT benchmark

The idea behind the Optimized Redundancy for Composite Algorithms applied to the FFT benchmark is to take advantage of the fact that the FFT can be easily decomposed into the sum over the even and odd indexes of the input signal, as shown in Equation 8.1:

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x(n) e^{-\frac{2\pi j}{N} nk} = \sum_{n=0}^{N-1} x(n) w_N^{nk} = \\ &= \sum_{n=0}^{N/2-1} x(2n) w_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1) w_N^{(2n+1)k} \end{aligned} \tag{8.1}$$

By the symmetry property of the twiddle factors, we also have Equation 8.2:

$$X_{k+N/2} = \sum_{n=0}^{N/2-1} x(2n) w_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1) w_N^{(2n+1)k} \tag{8.2}$$

This decomposition can be translated as computing two half-size FFT frames and performing a final butterfly operation:

$$\begin{aligned} X_k &= A_k + B_k \\ X_{k+N/2} &= A_k - B_k \end{aligned}$$

where:

$$A_k = \sum_{n=0}^{N/2-1} x(2n) w_N^{2nk}$$
$$B_k = \sum_{n=0}^{N/2-1} x(2n+1) w_N^{(2n+1)k}$$

The properties of A and B may vary depending on the characteristics of the input signal, but in general the information is distributed in a balanced manner between the even and odd input samples, so the magnitudes of A_k and B_k are similar. Thus, we can use the comparison between A_k and B_k to detect errors.

Fig. 8.2 shows the decomposition of an 8-point Radix-2 FFT into two 4-point FFTs. The first FFT uses the even samples and the second FFT uses the odd samples. The results are composed by means of a complex multiplication and a Butterfly Unit into the complete 8-point frame.

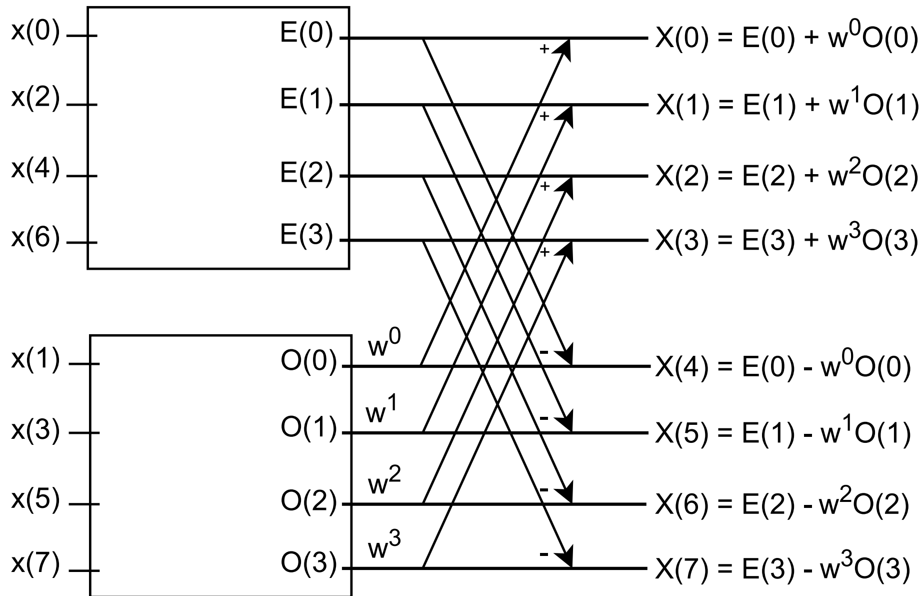


Figure 8.2: Diagram of an 8-point Fast Fourier Transform composed using two 4-point FFT blocks.

In our system, the last complex multiplication by the twiddle factors is embedded in the odd-sample FFT by modifying the twiddle factors in previous stages, and, thus, the composition step of the ORCA can be just implemented with a Butterfly Unit without multiplications.

Using the appropriate architectures for the reduced modules is important to take advantage of an ORCA design. If we use the same architecture for all the modules, the reduction of resources is minimal, since that way we are only eliminating the last stages of the reduced modules and substituting them with the composition block. On the other hand, as the reduced modules only must process half of the input data words, they can finish early. An optimal balance between area and performance can be obtained by

performing a serialization of the datapath in every two stages of the pipeline. This way, the number of stages is halved, to save resources, while the computing time is twice as long, but since they compute half the samples, the redundant modules match the performance of the full module.

In this work we have used the Radix-2 Single-path Delay Feedback (R2SDF) architecture [157], which we also used in the RRR FFT design of Chapter 6.

Fig. 8.3 illustrates the voting approach for the ORCA FFT. The reduced FFTs take, respectively, the even and odd samples. If the difference between the A and B components of the FFT exceeds the threshold, then we take the output from the F module. Otherwise, we select the output of the composite module obtained from the butterfly unit (G). Errors in the F module are not propagated because G, the output of the composite module, is taken by default. A difference between the reduced modules A and B larger than the threshold means that an error has affected one of them, in turn affecting the composed result G, so the F result has to be selected in that case. Only errors affecting the Least Significant Bits of A or B and produce a difference smaller than the threshold may be propagated to the output. We consider those as tolerable errors, since they produce small variations in the output.

This hardening approach works under the premise that particles produce single bit-flips and there is no error accumulation. However, because of the nature of SRAM-based FPGAs, the system may have to deal with errors in multiple parts of the circuit produced by single bit-flips, the so-called Common Mode Failures (CMF) [146]. CMFs affecting two or more of the triplicated devices produce an erroneous and uncorrectable result at the output of the voting logic. Additionally, if the fault affects the error detection and correction logic, a voting error, the other kind of uncorrectable error we consider, may arise at the output.

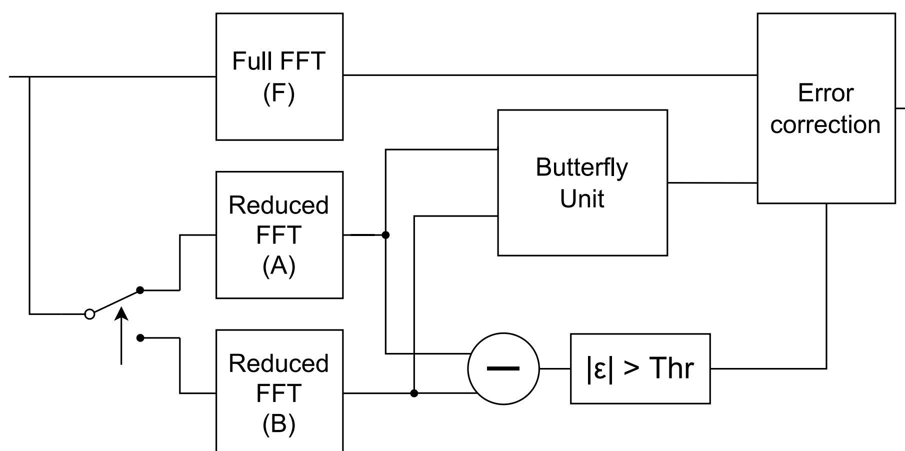


Figure 8.3: Block diagram of a Fast Fourier Transform benchmark hardened using the ORCA technique.

The voting logic as well as the final butterfly unit performing the composition of the reduced FFT modules, were hardened using a TMR approach to improve the hardness of the ORCA voter and minimize voting logic errors.

8.4.B. FIR filter benchmark

The Optimized Redundancy for Composite Algorithms hardening technique has been implemented for a FIR filter using a very similar approach as the one explained for the FFT.

A pipelined FIR filter can be implemented as the weighted sum of a certain amount of delayed inputs. The weighted sum may be broken down into smaller weighted sums of arbitrary dimensions and parallelized in hardware architectures to improve performance. In the proposed architecture, the sum is decomposed into its even and odd parts, as stated in Equation 8.3, where N is the order of the filter, c_k refers to the filter coefficients and x_k are the input samples of the filter.

$$y(n) = \sum_{k=0}^{N} c_k x_{n-k} = \sum_{k=0}^{N/2} c_{2k} x_{n-2k} + \sum_{k=0}^{N/2} c_{2k+1} x_{n-2k-1} \tag{8.3}$$

In terms of hardware, the basic FIR filter can be designed as in Fig. 8.4.a by decomposing the sums in two blocks, the one computing the even samples and the one computing the odd samples. This way, their results are composed again with a simple sum operation, as depicted in Fig. 8.4.b. The method to completely separate the two blocks in independent hardware modules is shown in Fig. 8.4.c.

As Fig. 8.4.c illustrates, a slight modification has to be introduced in the architecture of the reduced filters to ensure that the odd and even inputs are multiplied by their respective odd and even coefficients. For FPGA implementations, this modification may be easily implemented using the internal registers of the DSP blocks. In particular, this feature is supported in the Xilinx FIR compiler IP block [189] with the “Interpolated filter” option, which seamlessly adds a configurable number of delay registers between multiplications.

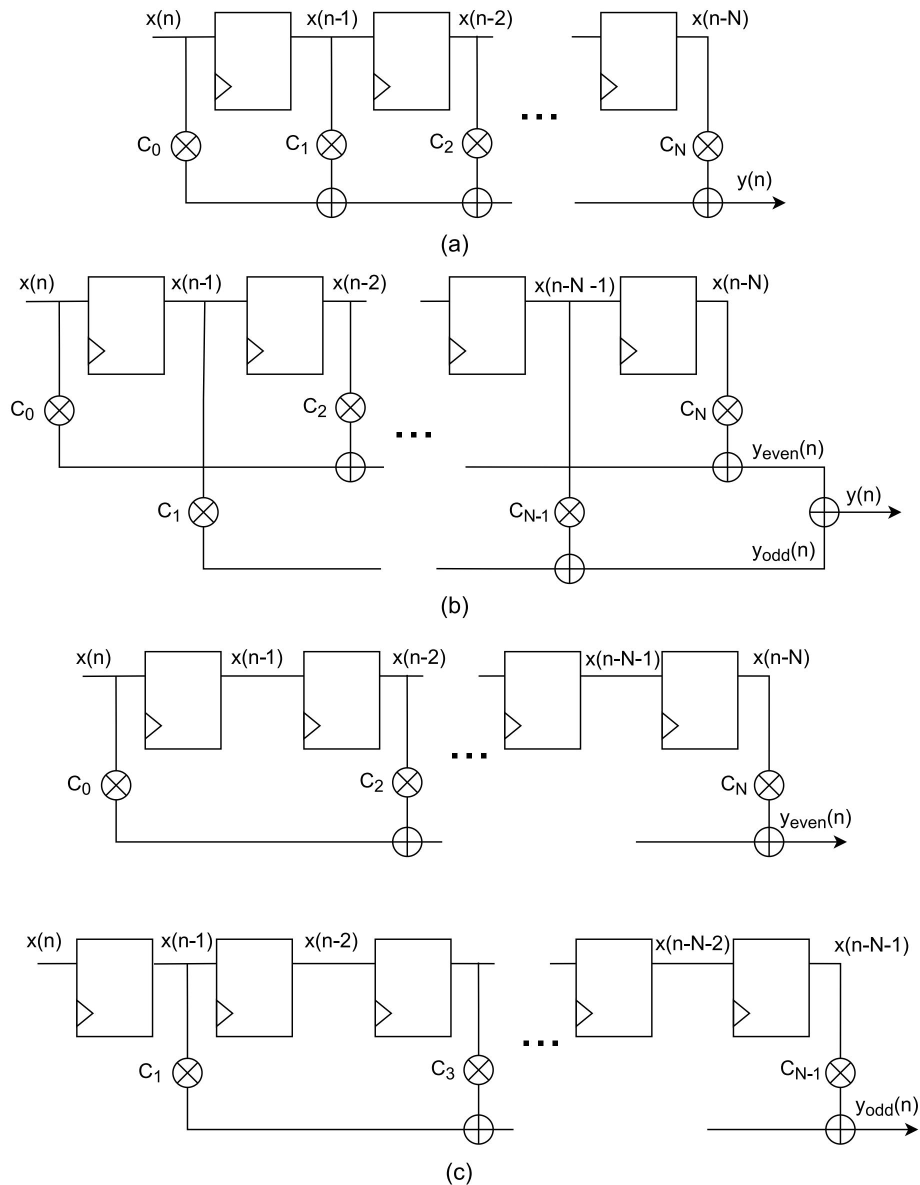


Figure 8.4: Finite Impulse Response filter decomposition in two blocks: (a) basic FIR filter architecture, (b) intermediate separation of the FIR filter, (c) final decomposition of the even and odd FIR filter components.

Fig. 8.5 represents the block design of the ORCA FIR filter. We have implemented two reduced FIR filters, each of them performing operations on half of the input samples, the even and the odd parts, respectively. The y_{even} and y_{odd} outputs are combined to obtain y_c, which should be exactly the same as y, the output of the full FIR filter, in the absence of errors. In case y and y_c differ, this means an error is present in any of the FIR blocks. The difference between y_{even} and y_{odd} is then used to discriminate where the error occurred. If this difference is greater than the threshold value, we conclude that the fault affected one of the reduced filters, but the result from F should be correct in that case. Otherwise, an error is present either in the F filter or in a reduced filter, as a small error. Anyway, the y_c result should be selected as it is the most correct output, yielding an exact correction or an approximate correction, with an error not greater than the threshold. CMFs and voting

logic errors can also affect the system, so we hardened the voting logic in a TMR fashion to reduce the sensitivity of the circuit to voting logic errors.

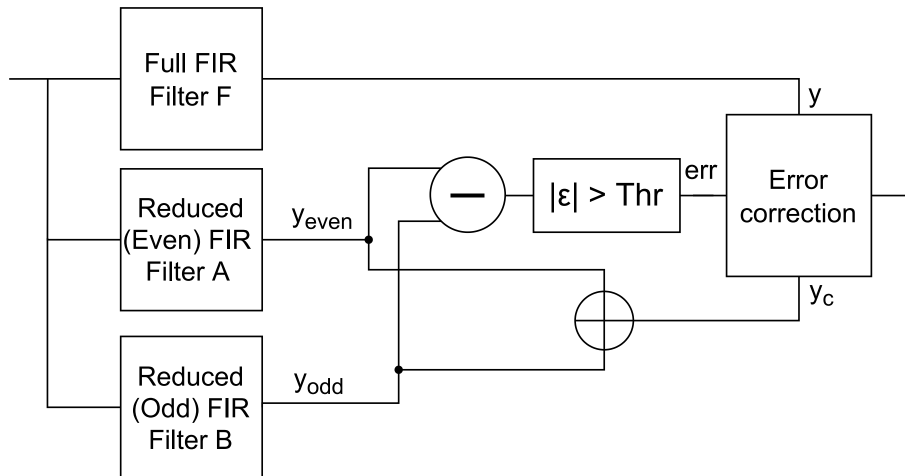


Figure 8.5: Block diagram of a Finite Impulse Response filter benchmark hardened with the ORCA technique.

8.5. Experimental setup

To evaluate the performance of the ORCA error mitigation technique we used fault injection. For the FFT benchmark, we hardened a Radix-2 SDF pipelined FFT by adding two redundant R2SER2 SDF FFTs. The Full FFT computes frames of 256 points, while the Reduced FFTs compute frames of 128 datapoints, alternating between the even and odd samples fed to the system.

As for the FIR filter implementation, the filters of the tested design were implemented using Xilinx FIR Compiler IP [189], which facilitates the implementation of highly customizable FIR filters of different types, orders and data widths. We implemented a low-pass 20-tap filter to be used as the Full filter, and two low-pass 10-tap filters with additional delay registers between multipliers to act as the two Reduced filters. A second benchmark using a 40-taps Full filter and two 20-taps Reduced filters was also tested to evaluate how the error rates scale with the use of resources. All the filter designs use 16-bit data with fixed-point operations.

Both the FFT and the FIR filter implementations have been wrapped in a testbench circuit, that generates the inputs and checks the outputs for errors. The results of the three blocks and the voted results are stored by the testbench and are compared them with a golden copy of the results. All of them are sent through a UART connection to an external host to be analyzed afterwards, in case of discrepancy with the golden data. Then, the external host resets and reconfigures the FPGA to avoid error accumulation.

We carried out the tests using the programmable logic of a Xilinx Zynq-7010 All Programmable System on Chip (APSoC) [138]. The ARM microprocessor embedded in this device was not used for the experiments.

Again, we used Xilinx Soft Error Mitigation (SEM) IP [74] to perform the error injection experiments. When a fault is injected, we let it progress before attempting the correction of the corrupted data in the configuration memory. This way, we can observe the impact of errors in the circuit and the performance of the hardening technique. The frequency of the fault injections was selected so that at most one fault was injected in each iteration.

8.6. Experimental results

Using the previosuly described setups we assessed the performance of the Optimized Redundancy for Composite Algorithms.

8.6.A. FFT benchmark

The synthesis results of the ORCA FFT are presented in Fig. 8.6 along with results for some of the other hardening techniques reviewed in this Thesis used to harden the same benchmark, namely TMR, RPR, Scaled RPR and RRR. In the resource consumption calculation we included the three FFT blocks, the triplicated voting logic and additional control logic related to the target blocks. When comparing the ORCA designs to TMR, we can see an overall reduction of resources. Compared to other Approximate Error Correction methods, we observe that the resources needed by an ORCA design are very similar to those needed by a Reduced Resolution Redundancy design and only slightly higher than those needed by the RPR implementation. However, the RPR design requires a higher number of DSPs because reducing the precision does not reduce the amount of operations. The scaled RPR, which is in between the TMR and the conventional RPR in terms of resources consumption, is also more consuming than ORCA.

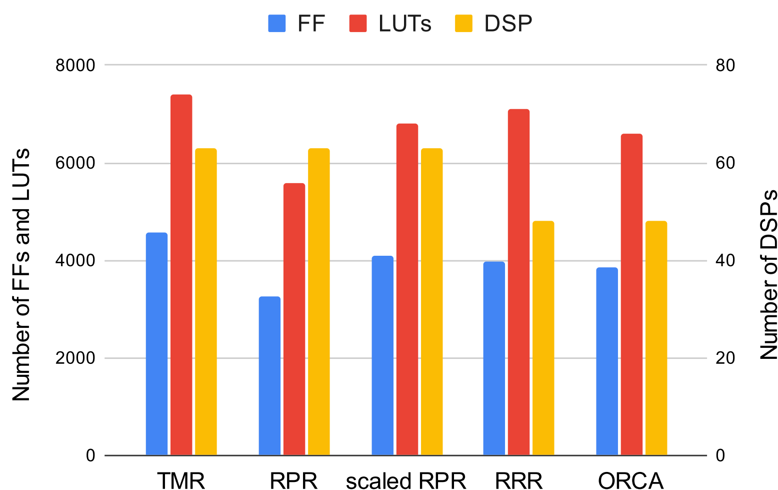


Figure 8.6: Synthesis results of the ORCA FFT benchmark and comparison with other hardening techniques applied to an R2SDF FFT.

Table 8.1 presents the results for the injection experiments performed in the FFT benchmark hardened using ORCA, TMR, Scaled RPR, conventional RPR and RRR.

In the first row of the table we show the number of addresses of the configuration memory of the FPGA in which faults where injected during the experiments. In the next two rows we present the total number of faulty FFT calculations and the amount of those that contain uncorrectable errors. To normalize these data and allow us to compare the various hardening methods, we calculated the error rate and the uncorrectable error rates, which are shown in the next two rows. Between parenthesis, we show the confidence intervals of those figures, calculated with a Gaussian distribution at 95% confidence.

The next four rows contain an analysis about the incidence of errors in the data words inside each faulty frame. The first of those lines contains the total number of words inside the faulty FFT calculations. In the second, we present the number and percentage of the total words containing a fault. In the last two rows we classify, in terms of percentage of the total, the erroneous words in two categories: correctable and uncorrectable errors. Correctable errors include errors that are either exactly corrected by the voter, corrected with an approximate result, or smaller than the threshold. Uncorrectable errors include Common Mode Failures (CMF) and errors in the voting logic.

The last row of Table 8.1 provides the Peak Signal-to-Noise Ratio, a measure of the disruption caused by the noise in the output of the circuit compared to the maximum value possibly taken by the signal. Higher PSNR values denote less noise in the output.

Table 8.1: RESULTS OF THE ORCA FFT HARDENING TECHNIQUE UNDER INJECTION EXPERIMENTS AND COMPARISON WITH OTHER HARDENING METHODS.

	TMR FFT	RPR FFT	Scaled RPR FFT	RRR FFT	ORCA FFT
Injected addresses ($\times10^5$)	3.6	3.6	3.6	3.6	3.6
Faulty frames	1649	1423	1565	1352	1400
Uncorrectable frames	51	66	47	14	23
Error rate ($\times10^{-3}$)	4.62 (4.4, 4.8)	3.95 (3.8, 4.2)	4.35 (4.1, 4.6)	3.76 (3.6, 4)	3.89 (3.7, 4.1)
Uncorrectable Error rate ($\times10^{-5}$)	14.28 (10.5, 18.6)	18.48 (14.2, 23.2)	13.05 (9.6, 17.4)	3.92 (2.1, 6.5)	6.39 (4.1, 9.6)
Total words in faulty frames	844288	728576	801280	692224	716800
Faulty words	267648 (31.7%)	133201 (18.3%)	159171 (19.9%)	155939 (22.5%)	145510 (20.3%)
Correctable (%)	93.6	93.5	91.3	98.8	98.5
Uncorrectable (%)	6.4	6.5	8.7	1.2	1.5
PSNR (dB)	77.3	79.9	86.9	65.8	81.9

From the results collected in Table 8.1, we observe that the ORCA FFT achieves a significant error sensitivity reduction, on a par with that of the RRR and the RPR. This error sensitivity reduction is strongly correlated with the reduction of resources achieved by the ORCA design. As we can see, all the Approximate Error Mitigation techniques shown in the table boast of lower error rates than the TMR implementation. The sensitivity of the ORCA FFT to uncorrectable errors is significantly lower than the other designs, except the RRR technique. However, we do not find a clear relationship between the uncorrectable error rate and the utilization of resources, leading to the conclusion that there are other factors at play when considering the sensitivity of a system

to CMF and voting logic errors. Size and distribution of the voting logic, as well as the placement of the FFTs and the architecture used by the designs may have a relevant impact on the appearance of uncorrectable errors.

The percentage of faulty words in the faulty frames is very similar in ORCA and the rest of the Approximate Error Mitigation techniques, and clearly lower than TMR. With respect to the uncorrectable faulty words, the ORCA and RRR designs are again much better than the rest. The percentage of uncorrectable data points found is below 2% of the total faulty data points, which is remarkably low and significantly smaller than in the RPR and TMR experiments.

In Fig. 8.7 we find a more in-depth analysis of the error classification of the faulty words found in the different experiments. This figure sheds additional light over the performance of the ORCA and helps understand the PSNR results. As explained in the theoretical analysis of the ORCA performed in Section III, existing Approximate Error Mitigation techniques correct errors in the full precision module with approximate results. However, designs hardened with the ORCA technique are able to perform exact corrections. As a consequence, ORCA shows the highest percentage of exact results. Moreover, the threshold used by the ORCA design is very small (0.15%), one order of magnitude smaller than the threshold used in the RPR design (1.5%). Therefore, the error in the tolerable case is much smaller.

Regarding the PSNR results, we observe that this metric is significantly high in the ORCA design. In particular, it is much higher than that of the RRR design, which is the worst of all cases. It is worth noting that this value is calculated only with the computations where some kind of noise reached the voted output. In the ORCA design this only encompasses tolerable and uncorrectable errors, whereas approximate corrections are also included in the PSNR values of the other designs. This high PSNR value is also a consequence of the smaller threshold used in the ORCA design.

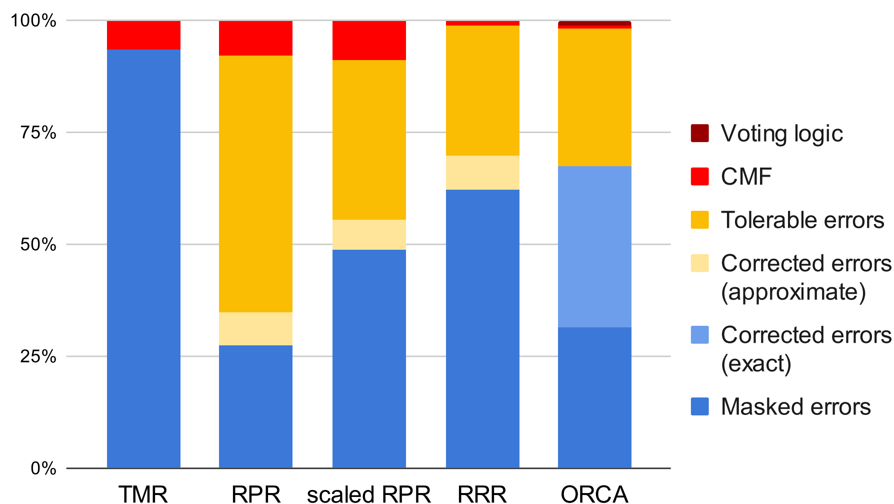


Figure 8.7: Error classification of the experiments conducted in the ORCA FFT benchmark and comparison with other hardening methods.

8.6.B. FIR filter benchmark

In Fig. 8.8 we present the synthesis results for the ORCA FIR filter as well as for the TMR and RPR hardening techniques applied to the same 20-tap FIR filters. Again, the ORCA technique is able to significantly reduce the number of DSPs, which is the most critical resource for this design.

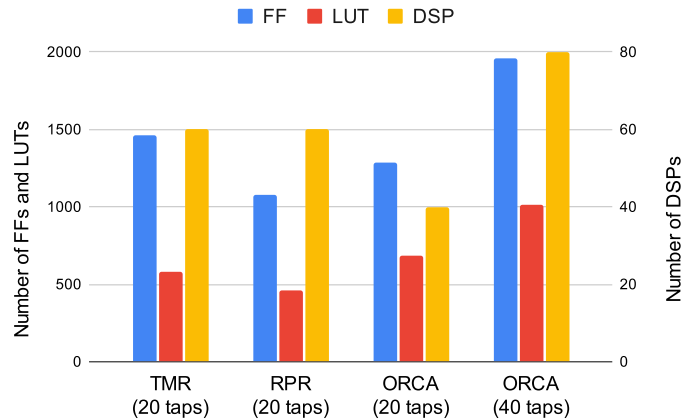


Figure 8.8: Synthesis results of the ORCA FIR filter benchmark and comparison with other hardening techniques applied to similar FIR filter implementations.

Table 8.2 shows the results of injection experiments performed on the FIR filter benchmarks. The results are presented in the same format as in the previous section. From the data in the table, we can see that the error sensitivity of the ORCA benchmark is lower than the TMR benchmark, and also slightly lower than in the RPR design. As seen on Fig. 8.8, the ORCA design offers a significant reduction of DSP blocks with respect to TMR and RPR, but the reduction is not so important in terms of FFs and LUTs. The FIR filters are mainly implemented using DSPs and this may explain the reduction of the error rate in the ORCA design. However, the logic inside the DSP block is less affected by the error injection, since only the configuration of the blocks is stored in the configuration memory an injection cannot be performed in the hardware inside the DSP blocks. As for the uncorrectable error rate, the ORCA shows a similar sensitivity to that of the RPR, the vast majority of the uncorrectable faults being related to errors in the voting process, which is more complex than the TMR voter. The effects of increasing the order of the filter in the ORCA design can be clearly seen in the table: the sensitivity of the circuit is increased due to the increase of resources, but the error correction capabilities only show a slight improvement.

Table 8.2: RESULTS OF THE ORCA FIR FILTER HARDENING TECHNIQUE UNDER INJECTION EXPERIMENTS AND COMPARISON WITH OTHER HARDENING METHODS.

	TMR FIR filter (20 taps)	**RPR FIR filter (20 taps)**	**ORCA FIR filter (20 taps)**	**ORCA FIR filter (40 taps)**
Injected addresses (×10^{5})	1.9	2.1	2	2
Faulty frames	395	355	309	548
Uncorrectable frames	4	21	25	54
Error rate (×10^{-3})	2.03 (1.87, 2.28)	1.64 (1.51, 1.87)	1.54 (1.37, 1.72)	2.74 (2.51, 2.97)
Uncorrectable Error rate (×10^{-5})	0.2 (0.06, 0.54)	0.97 (0.62, 1.53)	1.25 (0.81, 1.85)	2.69 (2.02, 3.52)
Total words in faulty frames	101120	90880	79104	140288
Faulty words	45784 (45.3%)	34080 (37.5%)	33289 (42.3%)	67802 (48.3%)
Correctable (%)	98.2	96	96.7	97.2
Uncorrectable (%)	1.8	4	3.3	2.8
PSNR (dB)	67.01	78.1	70.8	70.9

The error classification of the faulty words in the ORCA FIR filters is depicted in Fig. 8.9. Three aspects are notable when comparing the results. First, while CMFs had the highest percentage of uncorrectable errors in the TMR and RPR designs, voting logic errors contribute the most to the uncorrectable errors in the ORCA designs. This is probably due to the higher complexity of the voting logic in ORCA. Second, comparing the correctness of the results after the voting logic, we can observe that ORCA and RPR have around the same percentage of corrected errors, but in the ORCA implementation, those corrections are exact instead of approximate. This makes sense, since the architectures of those designs are very similar, but the voting process of the ORCA ensures better corrections. And third, increasing the order of the filter slightly improve the error correction capabilities of the ORCA designs even though the use of logic resources increases.

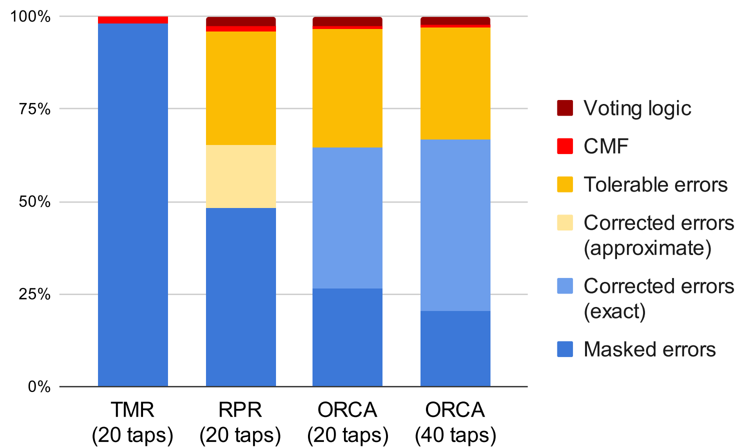


Figure 8.9: Error classification of the experiments conducted in the ORCA FIR filter benchmark and comparison with other hardening methods.

8.7. Conclusions

In this Chapter, we proposed a novel Approximate Error Mitigation technique specially conceived for composite algorithms implemented in FPGAs. In contrast to existing hardware redundancy techniques, the added redundant modules are complementary, not identical to the target design. Comparing the results of these complementary modules allows for error detection and their composition produces exact corrections, instead of the approximate results obtained in other previously reviewed hardening techniques. Thus, this technique is able to improve the probability of correcting errors with exact values.

The capabilities of the ORCA technique have been demonstrated with two typical Digital Signal Processing algorithms implemented in FPGA, but this is a very general approach and can be applied to a wide variety of composite algorithms. The results from the fault injection experiments we performed show that ORCA is able to improve the quality of the error mitigation results with lower overheads than other approximate techniques. From a general point of view, the ORCA hardening technique opens a new line of research regarding the utilization of complementary modules instead of identical modules in the implementation of hardware redundancy techniques for error mitigation.

9. CONCLUSIONS AND FUTURE RESEARCH

The main objective of this Thesis was the development and validation of Approximate Error Mitigation techniques for digital signal processing circuits in radiation environments.

At the beginning of this Thesis we validated the performance of the basic Reduced Precision Redundancy technique, which had not been tested under radiation at the time. We also proposed and evaluated a modified version of RPR that we called Scaled Reduced Precision Redundancy.

Pursuing the reduction of resources, and with the insights we obtained in those experiments, we proposed two novel Approximate Error Mitigation techniques, the Reduced Resolution Redundancy and the Optimized Redundancy for Composite Algorithms. These mitigation strategies can be applied to DSP algorithms that meet certain common requirements and they proved being worthy alternatives to the commonplace Triple Modular Redundancy mitigation technique.

Taking into account all the data from the experiments we conducted, the following main conclusions can be extracted:

- **Reduction of resources**

  Approximate mitigation techniques succeed in creating digital circuits hardened against radiation with minimal impact on the resources and power needed. The proposed approaches allow for on-the-fly error detection and correction, not just detection as is the case of DWC, while maintaining good error correction rates and low noise due to errors in the output of the hardened system, comparable to TMR techniques.

  Table 9.1 contains a numeric comparative of the overhead produced by some of the most common mitigation techniques and some of the designs proposed in this Thesis for 256-point, 32-bit Radix-2 Fast Fourier Transforms, compared to a non-hardened version of the circuit. The data presented in this table are intended to be a representative comparison between hardening techniques, although variations may exist due to differences in the hardened architectures or in the place and route.

Table 9.1: OVERHEAD INTRODUCED BY DIFFERENT HARDENING TECHNIQUES FOR A RADIX-2 FFT BENCHMARK

	FFT	TMR FFT	RPR FFT	Scaled RPR FFT	RRR FFT	ORCA FFT
Flip-Flop	1.0	2.8	2.0	2.5	2.4	2.6
LUT as logic	1.0	2.6	2.1	2.4	2.5	2.3
LUT as memory	1.0	2.9	1.9	2.5	2.7	2.7
DSP	1.0	3.0	3.0	3.0	2.0	2.0

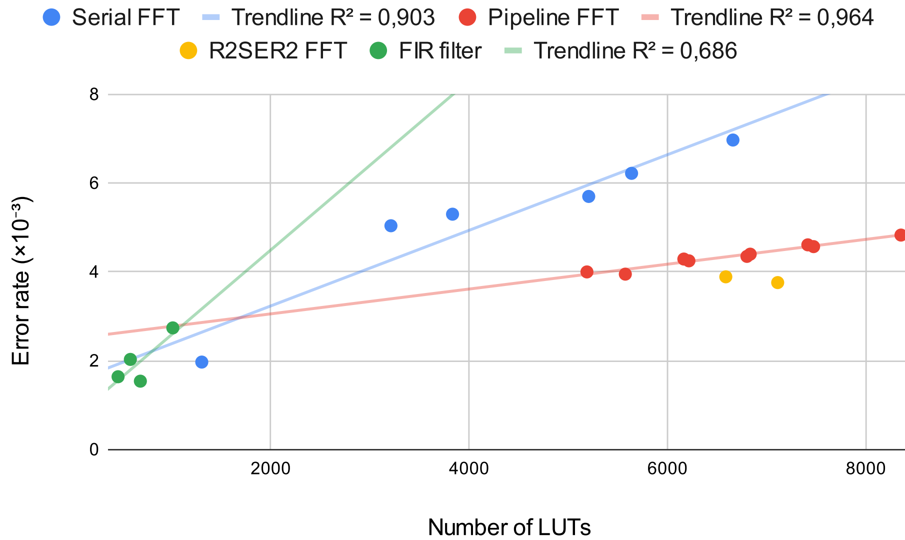
- **Fault injection sensitivity**
  Our fault injection experiments show that the error sensitivity of the circuit is directly linked to the amount, kind and proportion of resources used to implement the circuit. For the specific case of SRAM-based FPGAs, our results show a strong correlation between the amount of Look-Up Tables in the design and its error sensitivity.

  For the fault injection experiments, the correlation between the error sensitivity and other types of resources, i.e. flip-flops and DSP blocks, is not so evident. This may be explained because of the higher amount of configuration bits devoted to configure the behaviour of LUTs. The internal circuitry of FFs and DSPs is not affected by the fault injection, just their interfaces and interconnections, making them more resistant, and less influential, in the fault injection campaigns.

  An interesting finding related to the error sensitivity we found in this Thesis is the fact that the Radix-2, Radix-4 and Radix-2^2 FFT architectures employed in the RPR and TMR designs exhibit a very good correlation between the number of LUTs and their sensitivity, as we can see in Fig. 9.1a. We can also observe this trend in the experiments we performed with the FFTs using the serial Radix-4 architecture and in the experiments where the FIR filter benchmark was used. However, the FFT architecture used for the RRR and ORCA FFT replicas, the Radix-2-Serial-2 architecture, drastically changes the previous trend, reducing the sensitivity of both circuits. Fig. 9.1b shows the correlation between the number of LUTs and the sensitivity of the circuit for some of the benchmarks reviewed in this Thesis. This change in the behaviour could be attributed to the reduction of the pipeline stages necessary to compute the approximate result.

Pipeline FFTs
Trendline R^2 = 0,964
Error rate ($\times 10^{-3}$)
5,0
4,5
4,0
3,5
3,0
5500
6000
6500
7000
7500
8000
LUTs
R4 TMR
R2 TMR
R2² TMR
R4 RPR
R4 Scaled RPR
R2 Scaled RPR
R2² RPR
R2² Scaled RPR
R2 RPR

(a)



(b)

Figure 9.1: Correlation between the number of LUTs employed by the designs and their sensitivity.

- **Radiation sensitivity and correlation with fault injection**
  The correlation between the usage of LUTs and the cross section of the designs is not so strong in the radiation experiments. This deviation with respect to the fault injection results can be easily explained by radiation effects affecting resources in the circuit in which faults could not be injected. Besides provoking errors in the configuration and routing of flip-flops, DSPs and memories, ionizing particles can also affect the state of those components, adding new failure modes that did not exist in the fault injection experiments. The effect of the architecture of the design is even more evident in this metric. The R2SER2SDF-based designs distort the trend followed by the the other FFT designs. Fig. 9.2 shows the correlation between the number of LUTs and the cross section of each circuit. The RRR and the ORCA designs have been excluded for clarity.

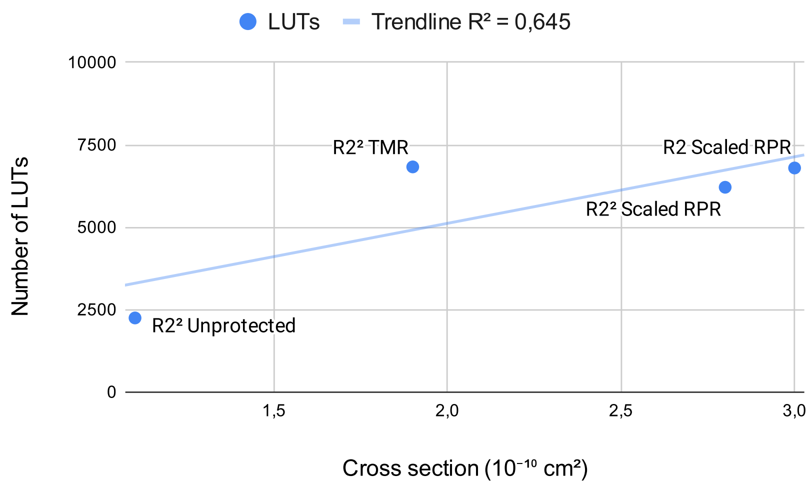


Figure 9.2: Correlation between the number of LUTs and the Cross section.

Nevertheless, the sensitivity of the designs under fault injection and radiation can be compared and our results suggest that the two methods may have comparable outputs. Fig. 9.3 shows that a correlation exists between the methods. This proves that faults in SRAM-based FPGAs are mainly driven by bit-upsets in the configuration memory and fault injection is a valid method to estimate the performance of a circuit exposed to radiation.

These results suggest that Approximate Error Mitigation techniques should be less sensitive than basic TMR, provided they are able to reduce the amount of resources needed by the hardened design.

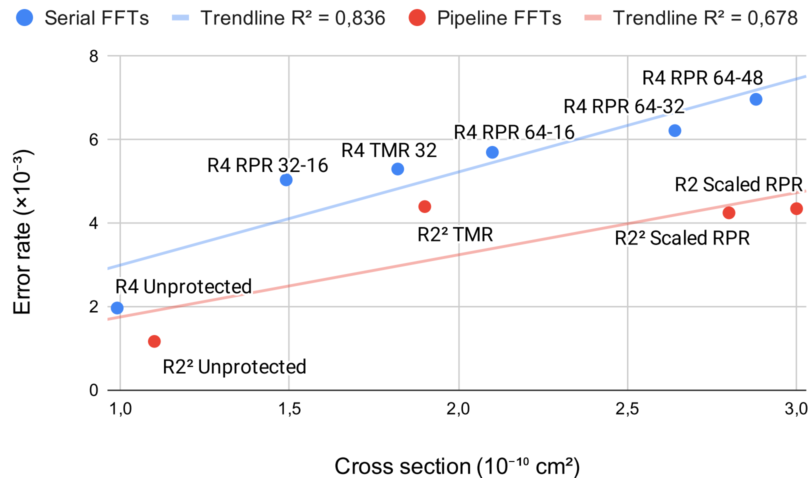


Figure 9.3: Correlation between the Error Rate and the Cross section.

- **Sensitivity to uncorrectable errors**

  No correlation can be appreciated in our results between the amount of any resources used by the designs and their sensitivity to CMFs, regardless of the particle or injection method used. As hinted by other researchers, the CMF cross section is not so much due to the amount of resources needed, but to the way they are laid out in the FPGA or ASIC circuit and whether there are interconnection points between the TMR domains. Thus, the differences in the CMF sensitivity of the proposed techniques are due to the architecture used in the design and not due to the resources employed or the Place and Route strategy, which remained default for every implementation.

  The proposed RRR and ORCA hardening techniques have shown excellent results in this regard, reducing the sensitivity to CMF of the designs hardened using them by up to three times in the case of RRR when compared to a TMR baseline design. There are two possible explanations to this effect, both related to the architecture used: first, due to the differences between the architectures of the modules, a data point may be processed in different clock cycles in the replicas than in the main module, causing the error to appear in different data points of the output frame, which may be corrected by the voting logic. And second, due to the decimation performed at the input of the

Reduced Resolution modules of the RRR, an error in the input data may be ignored, transforming a possible CMF into a simple error in the Full Resolution module, which may be corrected by the voter.

As for the other type of uncorrectable errors considered in this study, the complexity of the voting logic plays an obviously important role in the appearance of voting logic errors. In this case, the amount of resources used by the voter has a high influence on its sensitivity. Simplifying the voting blocks and protecting them using TMR to avoid single-point failures has been tested with satisfactory results in several of the designs developed. For instance, triplicating the voter in the RRR FFT design reduced the sensitivity of the circuit to errors in the voting logic by a factor of almost two.

- **Correction performance**
  Regarding the qualitative results of the proposed and analyzed Approximate Error Mitigation techniques, we have successfully determined that the performance upon error is generally good for all of them. The basic TMR technique offers exact corrections upon error, whereas the proposed techniques offer a range of approximate corrections that go from less precise corrections to exact corrections under certain circumstances and only approximate results in other cases.

  The average impact of errors in the voted output of the RPR, RRR and ORCA hardened benchmarks can be seen as just noise in the output of the circuit. Applications where noise may be tolerated can profit from an approximate hardening method. Approximate hardening solutions offer the unsupervised correction mechanism of a TMR with a lower resource cost, closer to that of a DWC mitigation, which is an attractive solution for low-budget projects.

  The quality of the results of an RPR-hardened module may be improved by using Scaled RPR in its implementation. Additionally, for those algorithms that allow it, the proposed ORCA mitigation technique is capable of performing exact corrections in a large fraction of the errors found, which greatly improves the quality of the corrections, as only a fraction of the corrections are approximate and even those produce a small noise in the output, as evidenced by the high PSNR value in the ORCA designs when compared to other error mitigation techniques.

- **Hardening heterogeneous devices**
  The analyses performed in the data extracted from the injection, but specially from the radiation experiments, showed us that all the components present in the Zynq devices are susceptible to failure. Errors in these components manifest themselves in different ways in the behaviour of the system, and some may even interfere in the correct functioning of the error detection and collection logic, altering the results. Some of these interferences can be easily found and discarded thanks to careful planning and redundancy techniques. However, some of the components involved in the data collection chain cannot be easily modified to allow error correction and we cannot be sure about the impact this kind of errors have in the presented results, although

they should improve were we able to protect all the components involved in the data collection.

The impact of the lack of hardening in components such as the Direct Memory Access controller or other data interfaces connecting the microprocessor with the FPGA is an obvious, but not very important, problem in our designs, where these resources do not play a mayor role. However, real applications making use of these resources intensively should consider error mitigation strategies specifically thought to detect and correct errors in data transfers between the microprocessor and the FPGA. Spatial redundancy hardening, with signatures and EDAC codes, mixed with temporal redundancy to recalculate results in case of discrepancy, could be used to produce RHBD mitigation to these errors.

The research conducted during this Thesis has shed additional light over the failure mechanisms of digital circuits implemented in SRAM-based FPGAs and how to mitigate them using standard and novel hardening techniques. The proposed techniques can be improved in several ways, which are the matter of future work:

- **Place and Route-oriented hardening**
  The incidence of uncorrectable errors is the weak point of any redundancy-based hardening technique. As explained, faults in the voting logic decisions can be decreased in number by simplifying and triplicating the logic. However, any N-Modular Redundancy architecture, including triplicated voters, is subject to CMFs. Future research regarding the incidence of uncorrectable errors should include the evaluation of different Place and Route techniques aimed at reducing CMFs on the proposed designs.

- **Combined hardening of MPSoCs**
  With the ultimate goal of providing a complete and systematic procedure to harden a Multiprocessor System-On-Chip, future research should include combining the proposed FPGA hardening methods with existing error mitigation strategies for microprocessors and explore different approaches to harden other relevant resources included in the device. The implementation of hardening techniques in the diverse components of complex and heterogeneous devices has been attempted separately in multiple occasions, but hardening complex applications that make use of all the resources available in the MPSoC remains off the beaten path and might be a challenging, yet interesting, work.

- **Other Approximate error mitigation techniques**
  The development of Approximate error mitigation techniques has proved to be a fruitful research line. Modifications and combinations of some of the presented designs, as well as new approaches to the concept, could be a promising starting point for future investigation in search for higher reduction of resources and better quality of the correction results.

BIBLIOGRAPHY

[1] L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, H. Martin, P. Martin-Holgado, and M. Garcia-Valderas, "Analyzing Reduced Precision Triple Modular Redundancy under Proton Irradiation," *IEEE Transactions on Nuclear Science*, vol. 69, pp. 470–477, 3 Mar. 2022. DOI: 10.1109/TNS.2022.3152088.

[2] L. Garcia-Astudillo, A. Lindoso, L. Entrena, H. Martín, and M. García-Valderas, "Error sensitivity study of FFT architectures implemented in FPGA," *Microelectronics Reliability*, vol. 126, p. 114 298, Nov. 2021. DOI: 10.1016/J.MICROREL.2021.114298.

[3] L. A. Garcia-Astudillo, A. Lindoso, L. Entrena, H. Martin, M. Garcia-Valderas, and P. Martin-Holgado, "Analyzing Scaled Reduced Precision Redundancy for Error Mitigation under Proton Irradiation," *IEEE Transactions on Nuclear Science*, pp. 1–1, 2022. DOI: 10.1109/TNS.2022.3147599.

[4] L. A. Garcia-Astudillo, L. Entrena, A. Lindoso, and H. Martin, "Reduced Resolution Redundancy: A Novel Approximate Error Mitigation Technique," *IEEE Access*, vol. 10, pp. 20 643–20 651, 2022. DOI: 10.1109/ACCESS.2022.3152202.

[5] L. A. Garcia-Astudillo *et al.*, "Evaluating Reduced Resolution Redundancy for radiation hardening," *IEEE Transactions on Nuclear Science*, 2023, (Early Access). DOI: 10.1109/TNS.2023.3268825.

[6] L. A. Garcia-Astudillo, A. Lindoso, and L. Entrena, "Error Mitigation using Optimized Redundancy for Composite Algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

[7] P. Aviles, L. Garcia-Astudillo, J. Belloch, L. Entrena, and A. Lindoso, "Comparative of proton radiation data for 28 nm Zynq-7000 SoC," *RADECS 2022 - European Conference on Radiation and its Effects on Components and Systems*, 2022.

[8] L. A. Garcia-Astudillo, A. Lindoso, M. Portela, and L. Entrena, "Evaluation of a Reduced Precision Redundancy FFT Design," *2020 35th Conference on Design of Circuits and Integrated Systems, DCIS 2020*, Nov. 2020. DOI: 10.1109/DCIS51330.2020.9268634.

[9] "Space engineering - Space Environment," European Cooperation for Space Standardization, Paris, France, Standard, Jun. 2020.

[10] W. H. Organization. "Ionizing radiation, health effects and protective measures." (2016), [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ionizing-radiation-health-effects-and-protective-measures (visited on 05/25/2022).

[11] W. Schimmerling, S. Curtis, L. B. Laboratory, and U. S. S. P. S. P. Office, *Workshop on the Radiation Environment of the Satellite Power System (SPS), Lawrence Berkeley Laboratory, Berkeley, California, September 15, 1978*, ser. CONF. Satellite Power System Project Office, 1980. [Online]. Available: https://books.google.es/books?id=gmWDugEACAAJ.

[12] P. V. Foukal, *Solar Astrophysics*. Wiley-VCH Verlag GmbH, 2013. [Online]. Available: https://books.google.es/books?id=z73VCHWjxjkC&pg=PA1&lpg=PA1&dq=sun+astrophysics&source=bl&ots=vte-oNikCe&sig=ACfU3U2--DZGoCXJZrKuKIPqUWrGy6xITQ&hl=es&sa=X&ved=2ahUKEwiLksSnpv33AhUFiv0HHZRQAKkQ6AF6BAgqEAM#v=onepage&q=sun%5C%20astrophysics&f=false.

[13] D. Besliu-Ionescu and M. Mierla, "Geoeffectiveness Prediction of CMEs," *Frontiers in Astronomy and Space Sciences*, vol. 8, p. 79, May 2021. DOI: 10.3389/FSPAS.2021.672203/BIBTEX.

[14] P. Vemareddy, "Successive injection of opposite magnetic helicity: Evidence for active regions without coronal mass ejections," *Monthly Notices of the Royal Astronomical Society*, vol. 507, pp. 6037–6044, 4 Sep. 2021. DOI: 10.1093/MNRAS/STAB2401. [Online]. Available: https://academic.oup.com/mnras/article/507/4/6037/6356577.

[15] M. Santos. "NASA Says Powerful Solar Winds Blew Mars' Water and Atmosphere Off the Planet." (2015), [Online]. Available: https://futurism.com/nasa-says-solar-storms-destroyed-mars-atmosphere-and-water (visited on 05/26/2022).

[16] "SPACECAST." (2017), [Online]. Available: https://fp7-spacecast.eu/ (visited on 05/26/2022).

[17] S. Gabici, "Gamma Ray Astronomy and the Origin of Galactic Cosmic Rays," Nov. 2008. DOI: 10.48550/arxiv.0811.0836. [Online]. Available: https://arxiv.org/abs/0811.0836v1.

[18] E. G. Berezhko, "Cosmic rays from active galactic nuclei," *Astrophys.J.Lett.*, vol. 684, pp. L69–L71, 2 Sep. 2008. DOI: 10.1086/592233.

[19] R. A. Mewaldt, "Galactic cosmic ray composition and energy spectra," *Advances in Space Research*, vol. 14, pp. 737–747, 10 Oct. 1994. DOI: 10.1016/0273-1177(94)90536-3.

[20] A. J. Dragt, M. M. Austin, and R. S. White, "Cosmic ray and solar proton albedo neutron decay injection," *Journal of Geophysical Research*, vol. 71, pp. 1293–1304, 5 Mar. 1966. DOI: 10.1029/JZ071I005P01293. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1029/JZ071i005p01293%20https://onlinelibrary.wiley.com/doi/abs/10.1029/JZ071i005p01293%20https://agupubs.onlinelibrary.wiley.com/doi/10.1029/JZ071i005p01293.

[21] A. M. Lenchek and S. F. Singer, "Geomagnetically trapped protons from cosmic-ray albedo neutrons," *Journal of Geophysical Research*, vol. 67, pp. 1263–1287, 4 Apr. 1962. DOI: 10.1029/JZ067I004P01263.

[22] "Cosmic Ray Albedo Neutron Decay (CRAND) as a Source of Inner Belt Electrons: Energy Spectrum Study," *Geophysical Research Letters*, vol. 46, pp. 544–552, 2 Jan. 2019. DOI: 10.1029/2018GL080887. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1029/2018GL080887%20https://onlinelibrary.wiley.com/doi/abs/10.1029/2018GL080887%20https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2018GL080887.

[23] M. J. Treadaway, B. C. Passenheinu, and B. D. Kitterer, "Luminescence and absorption of electron-irradiated common optical glasses, sapphire, and quartz," *IEEE Transactions on Nuclear Science*, vol. 22, pp. 2253–2258, 6 1975. DOI: 10.1109/TNS.1975.4328115.

[24] E. W. Taylor *et al.*, "Space radiation resistant hybrid and polymer materials for solar cells," *Conference Record of the IEEE Photovoltaic Specialists Conference*, pp. 2636–2641, 2010. DOI: 10.1109/PVSC.2010.5617191.

[25] J. R. Schwank *et al.*, "Radiation effects in MOS oxides," *IEEE Transactions on Nuclear Science*, vol. 55, pp. 1833–1853, 4 Aug. 2008. DOI: 10.1109/TNS.2008.2001040.

[26] R. L. Pease, R. D. Schrimpf, and D. M. Fleetwood, "ELDRS in Bipolar Linear Circuits: A Review," *IEEE Transactions on Nuclear Science*, vol. 56, no. 4, pp. 1894–1908, 2009. DOI: 10.1109/TNS.2008.2011485.

[27] "Space Product Assurance - Radiation Hardness Assurance - EEE components," European Cooperation for Space Standardization, Paris, France, Standard, Oct. 2012.

[28] D. Sinclair and J. Dyer, "Radiation effects and cots parts in smallsats," *Small Satellite Conference*, Aug. 2013. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2013/all2013/69 (visited on 07/16/2022).

[29] S. Buchner, *Radiation Hardness Assurance (RHA)*, Presented at International School on the Effects of Radiation on Embedded Systems for Space Applications (SERESSA) 2019, 2019.

[30] C. Poivey. "TNID Total Non Ionizing Dose or DD Displacement Damage." (2017), (visited on 12/10/2022).

[31] "Exploring the kinetics of formation and annealing of single particle displacement damage in microvolumes of silicon," *IEEE Transactions on Nuclear Science*, vol. 61, pp. 2826–2833, 6 Dec. 2014. DOI: 10.1109/TNS.2014.2364397.

[32] G. P. Summers, E. A. Burke, and M. A. Xapsos, "Displacement damage analogs to ionizing radiation effects," *Radiation Measurements*, vol. 24, pp. 1–8, 1 Jan. 1995. DOI: 10.1016/1350-4487(94)00093-G.

[33] C. Virmontois *et al.*, "Influence of displacement damage dose on dark current distributions of irradiated cmos image sensors," *Proceedings of the European Conference on Radiation and its Effects on Components and Systems, RADECS*, pp. 329–335, 2011. DOI: 10.1109/RADECS.2011.6131312.

[34] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Transactions on Device and Materials Reliability*, vol. 5, pp. 305–315, 3 Sep. 2005. DOI: 10.1109/TDMR.2005.853449.

[35] A. A. Keshavarz, T. A. Fischer, W. R. Dawes, and C. F. Hawkins, "Computer simulation of ionizing radiation† burnout in power mosfets," *IEEE Transactions on Nuclear Science*, vol. 35, pp. 1422–1427, 6 1988. DOI: 10.1109/23.25474.

[36] C. T. Dai and M. D. Ker, "Optimization of Guard Ring Structures to Improve Latchup Immunity in an 18 v DDDMOS Process," *IEEE Transactions on Electron Devices*, vol. 63, pp. 2449–2454, 6 Jun. 2016. DOI: 10.1109/TED.2016.2549598.

[37] R. Koga and W. A. Kolasinski, "Heavy ion induced snapback in CMOS devices," *IEEE Transactions on Nuclear Science*, vol. 36, pp. 2367–2374, 6 1989. DOI: 10.1109/23.45450.

[38] J. L. Titus, "An updated perspective of single event gate rupture and single event burnout in power MOSFETs," *IEEE Transactions on Nuclear Science*, vol. 60, pp. 1912–1928, 3 2013. DOI: 10.1109/TNS.2013.2252194.

[39] A. Haran, J. Barak, D. David, E. Keren, N. Refaeli, and S. Rapaport, "Single event hard errors in sram under heavy ion irradiation," *IEEE Transactions on Nuclear Science*, vol. 61, pp. 2702–2710, 5 Oct. 2014. DOI: 10.1109/TNS.2014.2345697.

[40] V. Ferlet-Cavrois, L. W. Massengill, and P. Gouker, "Single event transients in digital cmos - a review," *IEEE Transactions on Nuclear Science*, vol. 60, pp. 1767–1790, 3 2013. DOI: 10.1109/TNS.2013.2255624.

[41] "Space Product Assurance - Techniques for radiation effects mitigation in ASICs and FPGAs handbook," European Cooperation for Space Standardization, Paris, France, Standard, Jan. 2016.

[42] A. Perez-Celis and M. J. Wirthlin, "Statistical method to extract radiation-induced multiple-cell upsets in sram-based fpgas," *IEEE Transactions on Nuclear Science*, vol. 67, pp. 50–56, 1 Jan. 2020. DOI: 10.1109/TNS.2019.2955006.

[43] M. Sheehan, *The international politics of space*. Routledge, 2007.

[44] A. Golkar and A. Salado, "Definition of New Space—Expert Survey Results and Key Technology Trends," *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 2, pp. 2–9, 1 Dec. 2020. DOI: 10.1109/jmass.2020.3045851.

[45] I. Chatterjee, *Short course: Single-event effects - basic mechanisms and testing of complex devices*, RADECS 2021 Short Course, 2021. (visited on 08/03/2022).

[46] J. K. Ryan Brukardt and B. Stokes. "R&D for space: Who is actually funding it?" (2021), [Online]. Available: https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/r-and-d-for-space-who-is-actually-funding-it (visited on 08/03/2022).

[47] W. Lecky, "New Space and the role of public support. Part one.," European Space Agency, Tech. Rep., May 2016. [Online]. Available: https://esamultimedia.esa.int/docs/business_with_esa/New_Space_and_the_role_of_public_support_Part1.pdf.

[48] A. H. Sanchez, T. Soares, and A. Wolahan, "Reliability aspects of mega-constellation satellites and their impact on the space debris environment," Institute of Electrical and Electronics Engineers Inc., Mar. 2017. DOI: 10.1109/RAM.2017.7889671.

[49] "Space sustainability - Space debris mitigation requirements," European Cooperation for Space Standardization, Paris, France, Standard, Dec. 2019.

[50] "Space systems - Space debris mitigation requirements," ISO, Switzerland, Standard, Jul. 2019.

[51] "ECSS - Glossary of terms," European Cooperation for Space Standardization, Paris, France, Standard, Oct. 2012.

[52] "Space Product Assurance - Software dependability and safety," European Cooperation for Space Standardization, Paris, France, Standard, Nov. 2017.

[53] D. J. Wilkins. "The Bathtub Curve and Product Failure Behavior. Part One - The Bathtub Curve, Infant Mortality and Burn-in." (2002), [Online]. Available: https://www.weibull.com/hotwire/issue21/hottopics21.htm (visited on 07/26/2022).

[54] N. Rezzak, J. J. Wang, D. Dsilva, and N. Jat, "TID and SEE characterization of Microsemi's 4th generation radiation tolerant RTG4 flash-based FPGA," *IEEE Radiation Effects Data Workshop*, vol. 2015-November, Nov. 2015. DOI: 10.1109/REDW.2015.7336739.

[55] P. Ellerman, *Calculating Reliability using FIT & MTTF: Arrhenius HTOL Model*, English, version MicroNote 1002, Microsemi, 2009. [Online]. Available: https://www.microsemi.com/document-portal/doc_download/124041-calculating-reliability-using-fit-mttf-arrhenius-htol-model.

[56] "SPENVIS - Space Environment, Effects, and Education System." (), [Online]. Available: https://www.spenvis.oma.be/ (visited on 12/12/2022).

[57] "OMERE - Logiciel dédié à l'environnement spatial développé par TRAD." (), [Online]. Available: https://www.trad.fr/spatial/logiciel-omere/ (visited on 12/12/2022).

[58] J. A. Felix, P. E. Dodd, M. R. Shaneyfelt, J. R. Schwank, and G. L. Hash, "Radiation response and variability of advanced commercial foundry technologies," vol. 53, Dec. 2006, pp. 3187–3194. DOI: 10.1109/TNS.2006.886041.

[59] P. Martin-Holgado *et al.*, "How the Analysis of Archival Data Could Provide Helpful Information about TID Degradation. Case Study: Bipolar Transistors," *IEEE Transactions on Nuclear Science*, pp. 1–1, Jun. 2022. DOI: 10.1109/tns.2022.3185940.

[60] "Total Dose Steady-State Irradiation Test Method," European Space Components Coordination, The Netherlands, Standard, Jun. 2016.

[61] M. Poizat. "Total Ionizing Dose Testing." (2017), [Online]. Available: https://indico.cern.ch/event/635099/contributions/2570674/attachments/1456398/2249969/Radiation_Effects_and_RHA_ESA_Course_9-10_May_2017_TID_MP_FINAL_WIN.pdf (visited on 12/07/2022).

[62] "Guidelines for Displacement Damage Irradiation Testing," European Space Components Coordination, The Netherlands, Standard, Nov. 2019.

[63] "Single Event Effects Test Method and Guidelines," European Space Components Coordination, The Netherlands, Standard, Oct. 2014.

[64] "Standard Guide for the Measurement of Single Event Phenomena (SEP) Induced by Heavy Ion Irradiation of Semiconductor Devices," ASTM International, Pennsylvania, USA, Standard, Apr. 2018.

[65] "TEST PROCEDURE FOR THE MANAGEMENT OF SINGLE-EVENT EFFECTS IN SEMICONDUCTOR DEVICES FROM HEAVY ION IRRADIATION," JEDEC, Arlington, USA, Standard, Nov. 2017.

[66] "TEST STANDARD FOR THE MEASUREMENT OF PROTON RADIATION SINGLE EVENT EFFECTS IN ELECTRONIC DEVICES," JEDEC, Arlington, USA, Standard, Oct. 2013.

[67] "TEST METHOD FOR BEAM ACCELERATED SOFT ERROR RATE," JEDEC, Arlington, USA, Standard, Sep. 2021.

[68] "TEST METHOD STANDARD. ENVIRONMENTAL TEST METHODS FOR SEMICONDUCTOR DEVICES PART 1: TEST METHODS 1000 THROUGH 1999," Department of Defense, USA, Standard, Aug. 2016.

[69] "Evaluation report of 14-MeV neutron test methodology," RADSAGA, Switzerland, Standard, Jun. 2019.

[70] S. P. Buchner, F. Miller, V. Pouget, and D. P. McMorrow, "Pulsed-laser testing for single-event effects investigations," *IEEE Transactions on Nuclear Science*, vol. 60, pp. 1852–1875, 3 2013. DOI: 10.1109/TNS.2013.2255312.

[71] D. Nergui *et al.*, "Single-event transients in sige hbts induced by pulsed x-ray microbeam," *IEEE Transactions on Nuclear Science*, vol. 67, pp. 91–98, 1 Jan. 2020. DOI: 10.1109/TNS.2019.2959973.

[72] L. Entrena, "Fast fault injection techniques using fpgas," pp. 1–1, Jul. 2013. DOI: 10.1109/LATW.2013.6562680.

[73] L. Entrena, M. García-Valderas, R. Fernández-Cardenal, A. Lindoso, M. G. Portela, and C. López-Ongil, "Soft error sensitivity evaluation of microprocessors by multilevel emulation-based fault injection," *IEEE Transactions on Computers*, vol. 61, pp. 313–322, 3 2012. DOI: 10.1109/TC.2010.262.

[74] X. Inc., *Soft Error Mitigation Controller v4.1 LogiCORE IP Product Guide*, English, version 4.1, Xilinx Inc., 2022. [Online]. Available: https://docs.xilinx.com/r/en-US/pg036_sem.

[75] B. Z. - *et al.*, "Fault injection via on-chip debugging in the internal memory of systems-on-chip processor," *IOP Conference Series: Materials Science and Engineering*, vol. 94, p. 012 020, 1 Sep. 2015. DOI: 10.1088/1757-899X/94/1/012020. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/94/1/012020%20https://iopscience.iop.org/article/10.1088/1757-899X/94/1/012020/meta.

[76] M. Ebrahimi, A. Mohammadi, A. Ejlali, and S. G. Miremadi, "A fast, flexible, and easy-to-develop fpga-based fault injection technique," *Microelectronics Reliability*, vol. 54, pp. 1000–1008, 5 May 2014. DOI: 10.1016/J.MICROREL.2014.01.002.

[77] A. Coronetti *et al.*, "Radiation hardness assurance through system-level testing: Risk acceptance, facility requirements, test methodology, and data exploitation," *IEEE Transactions on Nuclear Science*, vol. 68, pp. 958–969, 5 May 2021. DOI: 10.1109/TNS.2021.3061197.

[78] M. Bucher *et al.*, "Total ionizing dose effects on analog performance of 65 nm bulk CMOS with enclosed-gate and standard layout," *IEEE International Conference on Microelectronic Test Structures*, vol. 2018-March, pp. 166–170, Jun. 2018. DOI: 10.1109/ICMTS.2018.8383790.

[79] J. Verbeeck, P. Leroux, and M. Steyaert, "Radiation effects upon the mismatch of identically laid out transistor pairs," in *2011 IEEE ICMTS International Conference on Microelectronic Test Structures*, 2011, pp. 194–197. DOI: 10.1109/ICMTS.2011.5976845.

[80] D. Gomez Toro, "Temporal Filtering with Soft Error Detection and Correction Technique for Radiation Hardening Based on a C-element and BICS," Ph.D. dissertation, Dec. 2014.

[81] T. Oldham, D. Chen, S. Buchner, and K. LaBel, *Radiation Effects Test Guideline Document for Nonvolatile Memories: Lessons Learned*, English, NASA Electronic Parts and Packaging (NEPP) Program, 2013. [Online]. Available: https://radhome.gsfc.nasa.gov/radhome/learned.htm.

[82] N. A. Dodds *et al.*, "Effectiveness of SEL hardening strategies and the latchup domino effect," *IEEE Transactions on Nuclear Science*, vol. 59, pp. 2642–2650, 6 2012. DOI: 10.1109/TNS.2012.2224374.

[83] P. Roche and G. Gasiot, "Impacts of front-end and middle-end process modifications on terrestrial soft error rate," *IEEE Transactions on Device and Materials Reliability*, vol. 5, pp. 382–395, 3 Sep. 2005. DOI: 10.1109/TDMR.2005.853451.

[84] C. Delepaut *et al.*, "LCL Current Control Loop Stability Design," *ESASP*, vol. 719, L. Ouwehand, Ed., p. 83, 2014. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2014ESASP.719E..83D/abstract.

[85] D. Selcan, G. Kirbis, and I. Kramberger, "Low Level Radiation and Fault Protection Techniques Suitable for Nanosatellite Missions," *2017 17th European Conference on Radiation and Its Effects on Components and Systems, RADECS 2017*, Jul. 2017. DOI: 10.1109/RADECS.2017.8696240.

[86] A. L. Sternberg *et al.*, "Effect of amplifier parameters on single-event transients in an inverting operational amplifier," *IEEE Transactions on Nuclear Science*, vol. 49 III, pp. 1496–1501, 3 Jun. 2002. DOI: 10.1109/TNS.2002.1039690.

[87] J. L. Andrews, J. E. Schroeder, B. L. Gingerich, W. A. Kolasinski, R. Koga, and S. E. Diehl, "Single event error immune CMOS ram," *IEEE Transactions on Nuclear Science*, vol. 29, pp. 2040–2043, 6 1982. DOI: 10.1109/TNS.1982.4336492.

[88] O. A. Amusan, L. W. Massengill, B. L. Bhuva, S. DasGupta, A. F. Witulski, and J. R. Ahlbin, "Design techniques to reduce SET pulse widths in deep-submicron combinational logic," *IEEE Transactions on Nuclear Science*, vol. 54, pp. 2060–2064, 6 Dec. 2007. DOI: 10.1109/TNS.2007.907754.

[89] J. Ragnar Ahlbin, "Characterization of the mechanisms affecting Single Event Transients in sub-100 nm technologies.," Ph.D. dissertation, May 2012, p. 132.

[90] S. E. Armstrong, B. D. Olson, W. T. Holman, J. Warner, D. McMorrow, and L. W. Massengill, "Demonstration of a Differential Layout Solution for Improved ASET Tolerance in CMOS A/MS Circuits," *IEEE Transactions on Nuclear Science*, Dec. 2010. DOI: 10.1109/TNS.2010.2080320. [Online]. Available: http://ieeexplore.ieee.org/document/5658059/.

[91] J. R. Ahlbin, L. W. Massengill, B. L. Bhuva, B. Narasimham, M. J. Gadlage, and P. H. Eaton, "Single-event transient pulse quenching in advanced CMOS logic circuits," *IEEE Transactions on Nuclear Science*, vol. 56, pp. 3050–3056, 6 Dec. 2009. DOI: 10.1109/TNS.2009.2033689.

[92] T. D. Loveless, L. W. Massengill, B. L. Bhuva, W. T. Holman, A. F. Witulski, and Y. Boulghassoul, "A hardened-by-design technique for RF digital phase-locked loops," *IEEE Transactions on Nuclear Science*, vol. 53, pp. 3432–3438, 6 Dec. 2006. DOI: 10.1109/TNS.2006.886203.

[93] J. Teifel, "Self-voting dual-modular-redundancy circuits for single-event-transient mitigation," *IEEE Transactions on Nuclear Science*, vol. 55, pp. 3435–3439, 6 Dec. 2008. DOI: 10.1109/TNS.2008.2005583.

[94] M. Berg, K. LaBel, J. P. .-. N. E. Parts, Packaging, and undefined 2015, "Single event effects in FPGA devices 2014-2015," *ntrs.nasa.gov*, 2015. [Online]. Available: https://ntrs.nasa.gov/api/citations/20150015964/downloads/20150015964.pdf.

[95] L. A. Tambara, F. L. Kastensmidt, P. Rech, and C. Frost, "Decreasing FIT with diverse triple modular redundancy in SRAM-based FPGAs," *Proceedings - IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 153–158, Nov. 2014. DOI: 10.1109/DFT.2014.6962070.

[96] A. J. Sanchez-Clemente, "Transient error mitigation by means of approximate logic circuits.," Ph.D. dissertation, Jul. 2017, p. 218.

[97] B. Shim and N. R. Shanbhag, "Reduced precision redundancy for low-power digital filtering," *Conference Record of the Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 148–152, 2001. DOI: 10.1109/ACSSC.2001.986896.

[98] B. Shim and N. R. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 336–348, 4 Apr. 2006. DOI: 10.1109/TVLSI.2006.874359.

[99] P. S. Krishna and M. Vinodhini, "Performance analysis of different reduced precision redundancy based full adders," *2020 IEEE International Conference for Innovation in Technology, INOCON 2020*, Nov. 2020. DOI: 10.1109/INOCON50539.2020.9298240.

[100] M. A. Sullivan, H. H. Loomis, and A. A. Ross, "Employment of reduced precision redundancy for fault tolerant fpga applications," *Proceedings - IEEE Symposium on Field Programmable Custom Computing Machines, FCCM 2009*, pp. 283–286, 2009. DOI: 10.1109/FCCM.2009.53.

[101] I. C. Wey, C. C. Peng, and F. Y. Liao, "Reliable low-power multiplier design using fixed-width replica redundancy block," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, pp. 78–87, 1 Jan. 2015. DOI: 10.1109/TVLSI.2014.2303487.

[102] P. Reviriego, C. Bleakley, J. A. Maestro, and A. O'Donnell, "Offset DMR: A low overhead soft error detection and correction technique for transform-based convolution," *IEEE Transactions on Computers*, vol. 60, pp. 1511–1516, 10 2011. DOI: 10.1109/TC.2011.80.

[103] B. Pratt, M. Fuller, M. Rice, and M. Wirthlin, "Reduced-precision redundancy for reliable FPGA communications systems in high-radiation environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, pp. 369–380, 1 2013. DOI: 10.1109/TAES.2013.6404109.

[104] A. Gavros, H. H. Loomis, and A. A. Ross, "Reduced precision redundancy in a radix-4 fft implementation on a field programmable gate array," *IEEE Aerospace Conference Proceedings*, 2011. DOI: 10.1109/AERO.2011.5747459.

[105] W. Liu, Q. Liao, F. Qiao, W. Xia, C. Wang, and F. Lombardi, "Approximate Designs for Fast Fourier Transform (FFT) with Application to Speech Recognition," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, pp. 4727–4739, 12 Dec. 2019. DOI: 10.1109/TCSI.2019.2933321.

[106] F. F. D. Santos *et al.*, "Reduced Precision DWC: An Efficient Hardening Strategy for Mixed-Precision Architectures," *IEEE Transactions on Computers*, vol. 71, pp. 573–586, 3 Mar. 2022. DOI: 10.1109/TC.2021.3058872.

[107] X. Inc., *Radiation-Hardened, Space-Grade Virtex-5QV Family Data Sheet: Overview*, English, version 1.6, Xilinx Inc., 2018, 17 pp. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ds192_V5QV_Device_Overview.

[108] M. Dong, W. Pan, Z. Qiu, X. Qi, L. Zheng, and H. Liu, "A universal, low-delay, sec-dec-taec code for state register protection," *IEEE Access*, vol. 10, pp. 57 665–57 673, 2022. DOI: 10.1109/ACCESS.2022.3178953.

[109] Y. Li, S. Yao, J. Xu, and J. Gao, "A self-checking approach for seu/mbus-hardened fsms design based on the replication of one-hot code," *IEEE Transactions on Nuclear Science*, vol. 59, pp. 2572–2579, 5 PART 3 2012. DOI: 10.1109/TNS.2012.2212209.

[110] S. Chellappa, L. Clark, K. Holbert, Y. Cao, and U. Ogras, "Radiation hardened clock design," Ph.D. dissertation, 2015.

[111] S. E. Diehl, A. Ochoa, P. V. Dressendorfer, R. Koga, and W. A. Kolasinski, "Error analysis and prevention of cosmic ion-induced soft errors in static cmos rams," *IEEE Transactions on Nuclear Science*, vol. 29, pp. 2032–2039, 6 1982. DOI: 10.1109/TNS.1982.4336491.

[112] Y. Shiyanovskii, F. Wolff, and C. Papachristou, "Sram cell design protected from seu upsets," *Proceedings - 14th IEEE International On-Line Testing Symposium, IOLTS 2008*, pp. 169–170, 2008. DOI: 10.1109/IOLTS.2008.49.

[113] D. Bessot and R. Velazco, "Design of seu-hardened cmos memory cells: The hit cell," pp. 563–570, Dec. 2002. DOI: 10.1109/RADECS.1993.316519.

[114] T. Câlin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron cmos technology," *IEEE Transactions on Nuclear Science*, vol. 43, pp. 2874–2878, 6 PART 1 1996. DOI: 10.1109/23.556880.

[115] E. J. W. Wesley Peterson, *Error Correcting Codes*. 1961. [Online]. Available: https://books.google.es/books?hl=es&lr=%5C&id=5kfwlFeklx0C%5C&oi=fnd%5C&pg=PA1%5C&dq=error+correcting+codes%5C&ots=P-GbG8idLF&sig=ECTaY8EoAMuAPqGHHnF-J4xxkeA#v=onepage%5C&q=error%5C%20correcting%5C%20codes%5C&f=false.

[116] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147–160, 2 1950. DOI: 10.1002/J.1538-7305.1950.TB00463.X.

[117] Y. Li, B. Nelson, and M. Wirthlin, "Reliability models for sec/ded memory with scrubbing in fpga-based designs," *IEEE Transactions on Nuclear Science*, vol. 60, pp. 2720–2727, 4 2013. DOI: 10.1109/TNS.2013.2251902.

[118] G. He, S. Zheng, and N. Jing, "A hierarchical scrubbing technique for seu mitigation on sram-based fpgas," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, pp. 2134–2145, 10 Oct. 2020. DOI: 10.1109/TVLSI.2020.3010647.

[119] M. Wirthlin, D. Lee, G. Swift, and H. Quinn, "A method and case study on identifying physically adjacent multiple-cell upsets using 28-nm, interleaved and secded-protected arrays," *IEEE Transactions on Nuclear Science*, vol. 61, pp. 3080–3087, 6 Dec. 2014. DOI: 10.1109/TNS.2014.2366913.

[120] F. L. Kastensmidt, E. C. P. Fonseca, R. G. Vaz, O. L. Goncalez, R. Chipana, and G. I. Wirth, "TID in Flash-Based FPGA: Power Supply-Current Rise and Logic Function Mapping Effects in Propagation-Delay Degradation," *IEEE Transactions on Nuclear Science*, vol. 58, no. 4, pp. 1927–1934, 2011. DOI: 10.1109/TNS.2011.2128881.

[121] "Single Event Effects - A Comparison of Configuration Upsets and Data Upsets," Microsemi, USA, White paper, Nov. 2015. [Online]. Available: https://www.microsemi.com/document-portal/doc_view/135837-wp0203-single-event-effects-a-comparison-of-configuration-upsets-and-data-upsets.

[122] L. Sterpone and M. Violante, "A new reliability-oriented place and route algorithm for SRAM-based FPGAs," *IEEE Transactions on Computers*, vol. 55, pp. 732–744, 6 Jun. 2006. DOI: 10.1109/TC.2006.82.

[123] M. J. Cannon, A. M. Keller, H. C. Rowberry, C. A. Thurlow, A. Perez-Celis, and M. J. Wirthlin, "Strategies for Removing Common Mode Failures from TMR Designs Deployed on SRAM FPGAs," *IEEE Transactions on Nuclear Science*, vol. 66, pp. 207–215, 1 Jan. 2019. DOI: 10.1109/TNS.2018.2877579.

[124] R. Zhang, L. Xiao, J. Li, X. Cao, and L. Li, "An adjustable and fast error repair scrubbing method based on xilinx essential bits technology for sram-based fpga," *IEEE Transactions on Reliability*, vol. 69, pp. 430–439, 2 Jun. 2020. DOI: 10.1109/TR.2019.2896897.

[125] J. Heiner, B. Sellers, M. Wirthlin, and J. Kalb, "Fpga partial reconfiguration via configuration scrubbing," *FPL 09: 19th International Conference on Field Programmable Logic and Applications*, pp. 99–104, 2009. DOI: 10.1109/FPL.2009.5272543.

[126] M. Pignol, "Dmt and dt2: Two fault-tolerant architectures developed by cnes for cots-based spacecraft supercomputers," *Proceedings - IOLTS 2006: 12th IEEE International On-Line Testing Symposium*, vol. 2006, pp. 203–212, 2006. DOI: 10.1109/IOLTS.2006.24.

[127] M. Pignol, *System hardening and real space applications*, Presented at International School on the Effects of Radiation on Embedded Systems for Space Applications (SERESSA) 2019, 2019.

[128] M. Pena-Fernandez *et al.*, "Hybrid lockstep technique for soft error mitigation," *IEEE Transactions on Nuclear Science*, vol. 69, pp. 1574–1581, 7 Jul. 2022. DOI: 10.1109/TNS.2022.3149867.

[129] A. B. D. Oliveira *et al.*, "Lockstep dual-core arm a9: Implementation and resilience analysis under heavy ion-induced soft errors," *IEEE Transactions on Nuclear Science*, vol. 65, pp. 1783–1790, 8 Aug. 2018. DOI: 10.1109/TNS.2018.2852606.

[130] M. Violante, C. Meinhardt, R. Reis, and M. S. Reorda, "A low-cost solution for deploying processor cores in harsh environments," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 2617–2626, 7 Jul. 2011. DOI: 10.1109/TIE.2011.2134054.

[131] H. M. Pham, S. Pillement, and S. J. Piestrak, "Low-overhead fault-tolerance technique for a dynamically reconfigurable softcore processor," *IEEE Transactions on Computers*, vol. 62, pp. 1179–1192, 6 2013. DOI: 10.1109/TC.2012.55.

[132] O. Goloubeva, M. Rebaudengo, M. S. Reorda, and M. Violante, *Software-Implemented hardware Fault Tolerance*. 2006. [Online]. Available: https://link.springer.com/book/10.1007/0-387-32937-4.

[133] D. Sabogal and A. D. George, "Towards resilient spaceflight systems with virtualization," *IEEE Aerospace Conference Proceedings*, vol. 2018-March, pp. 1–8, Jun. 2018. DOI: 10.1109/AERO.2018.8396689.

[134] A. Agrawal *et al.*, "Approximate computing: Challenges and opportunities," Institute of Electrical and Electronics Engineers Inc., Nov. 2016. DOI: 10.1109/ICRC.2016.7738674.

[135] A. Aponte-Moreno, C. Pedraza, and F. Restrepo-Calle, "Reducing overheads in software-based fault tolerant systems using approximate computing," Institute of Electrical and Electronics Engineers Inc., May 2019. DOI: 10.1109/LATW.2019.8704586.

[136] H. J. Wunderlich, C. Braun, and A. Schöll, "Fault tolerance of approximate compute algorithms," vol. 2016-May, IEEE Computer Society, May 2016. DOI: 10.1109/VTS.2016.7477307.

[137] C. Oliveira *et al.*, "Validation of an on-chip watchdog for embedded systems exposed to radiation and conducted emi," *2014 9th Southern Conference on Programmable Logic, SPL 2014*, Jan. 2015. DOI: 10.1109/SPL.2014.7002212.

[138] X. Inc., *Zynq-7000 SoC Data Sheet: Overview*, English, version 1.11.1, Xilinx Inc., 2018, 25 pp. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ds190-Zynq-7000-Overview.

[139] X. Inc., *Zynq UltraScale+ MPSoC Data sheet: Overview*, English, version 1.10, Xilinx Inc., 2022, 42 pp. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ds891-zynq-ultrascale-plus-overview.

[140] X. Inc., *Versal Architecture and Product Data Sheet: Overview*, English, version 1.17, Xilinx Inc., 2022, 35 pp. [Online]. Available: https://www.xilinx.com/content/dam/xilinx/support/documents/data_sheets/ds950-versal-overview.pdf.

[141] L. Parra *et al.*, "A new hybrid nonintrusive error-detection technique using dual control-flow monitoring," *IEEE Transactions on Nuclear Science*, vol. 61, pp. 3236–3243, 6 Dec. 2014. DOI: 10.1109/TNS.2014.2361953.

[142] M. Pena-Fernandez, A. Lindoso, L. Entrena, M. Garcia-Valderas, Y. Morilla, and P. Martin-Holgado, "Online error detection through trace infrastructure in arm microprocessors," *IEEE Transactions on Nuclear Science*, vol. 66, pp. 1457–1464, 7 Jul. 2019. DOI: 10.1109/TNS.2019.2921767.

[143] P. M. Aviles, A. Lindoso, J. A. Belloch, M. Garcia-Valderas, Y. Morilla, and L. Entrena, "Radiation testing of a multiprocessor macrosynchronized lockstep architecture with freertos," *IEEE Transactions on Nuclear Science*, vol. 69, pp. 462–469, 3 Mar. 2022. DOI: 10.1109/TNS.2021.3129164.

[144] P. M. Aviles, L. Schäfer, A. Lindoso, J. A. Belloch, and L. Entrena, "High complexity reliable space applications in commercial microprocessors," *Microelectronics Reliability*, vol. 138, p. 114 679, Nov. 2022. DOI: 10.1016/j.microrel.2022.114679.

[145] K. S. Morgan, D. L. McMurtrey, B. H. Pratt, and M. J. Wirthlin, "A comparison of TMR with alternative fault-tolerant design techniques for FPGAs," *IEEE Transactions on Nuclear Science*, vol. 54, pp. 2065–2072, 6 Dec. 2007. DOI: 10.1109/TNS.2007.910871.

[146] H. Quinn, K. Morgan, P. Graham, J. Krone, M. Caffrey, and K. Lundgreen, "Domain crossing errors: Limitations on single device triple-modular redundancy circuits in xilinx FPGAs," *IEEE Transactions on Nuclear Science*, vol. 54, pp. 2037–2043, 6 Dec. 2007. DOI: 10.1109/TNS.2007.910870.

[147] J. Y. Jou and J. A. Abraham, “Fault-Tolerant Matrix Arithmetic and Signal Processing on Highly Concurrent Computing Structures,” *Proceedings of the IEEE*, vol. 74, pp. 732–741, 5 1986. DOI: 10.1109/PROC.1986.13535.

[148] T. H. Chen and L. G. Chen, “Concurrent Error-Detectable Butterfly Chip for Real-Time FFT Processing Through Time Redundancy,” *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 537–547, 5 1993. DOI: 10.1109/4.229402.

[149] J. Y. Jou and J. A. Abraham, “Fault-Tolerant FFT Networks,” *IEEE Transactions on Computers*, vol. 37, pp. 548–561, 5 1988. DOI: 10.1109/12.4606.

[150] F. Lombardi and J. C. Muzio, “Concurrent Error Detection and Fault Location in an FFT Architecture,” *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 728–736, 5 1992. DOI: 10.1109/4.133159.

[151] J. F. Li, S. K. Lu, S. A. Hwang, and C. W. Wu, “Easily testable and fault-tolerant FFT butterfly networks,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, pp. 919–929, 9 Sep. 2000. DOI: 10.1109/82.868460.

[152] A. Antola, R. Negrini, M. G. Sami, and N. Scarabottolo, “Fault-tolerance in FFT arrays: Time-redundancy approaches,” *Conference Record - International Conference on Communications*, vol. 3, pp. 779–785, 1990. DOI: 10.1109/ICC.1990.117182.

[153] Y. M. Hsu and E. E. Swartzlander, “FFT arrays with built-in error correction,” *Conference Record - Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 172–176, 1994. DOI: 10.1109/ACSSC.1994.471439.

[154] K. H. Huang and J. A. Abraham, “Algorithm-Based Fault Tolerance for Matrix Operations,” *IEEE Transactions on Computers*, vol. C-33, pp. 518–528, 6 1984. DOI: 10.1109/TC.1984.1676475.

[155] F. L. Kastensmidt, L. Sterpone, L. Carro, and M. S. Reorda, “On the optimal design of triple modular redundancy logic for SRAM-based FPGAs,” *Proceedings -Design, Automation and Test in Europe, DATE ’05*, vol. II, pp. 1290–1295, 2005. DOI: 10.1109/DATE.2005.229.

[156] S. He and M. Torkelson, “Designing pipeline FFT processor for OFDM (de) modulation,” *Conference Proceedings of the International Symposium on Signals, Systems and Electronics*, pp. 257–262, 1998. DOI: 10.1109/issse.1998.738077.

[157] E. H. Wold and A. M. Despain, “Pipeline and Parallel-Pipeline FFT Processors for VLSI Implementations,” *IEEE Transactions on Computers*, vol. C-33, pp. 414–426, 5 1984. DOI: 10.1109/TC.1984.1676458.

[158] D. I. Tao and C. R. Hartmann, “A Novel Concurrent Error Detection Scheme for FFT Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, pp. 198–221, 2 1993. DOI: 10.1109/71.207595.

[159] C. G. Oh and H. Y. Youn, "On Concurrent Error Location and Correction of FFT Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, pp. 257–260, 2 1994. DOI: 10.1109/92.285753.

[160] M. R. Choudhury and K. Mohanram, "Low cost concurrent error masking using approximate logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, pp. 1163–1176, 8 2013. DOI: 10.1109/TCAD.2013.2250581.

[161] A. J. Sanchez-Clemente, L. Entrena, R. Hrbacek, and L. Sekanina, "Error mitigation using approximate logic circuits: A comparison of probabilistic and evolutionary approaches," *IEEE Transactions on Reliability*, vol. 65, pp. 1871–1883, 4 Dec. 2016. DOI: 10.1109/TR.2016.2604918.

[162] A. Sánchez, L. Entrena, and F. Kastensmidt, "Approximate TMR for selective error mitigation in FPGAs based on testability analysis," *2018 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2018*, pp. 112–119, Nov. 2018. DOI: 10.1109/AHS.2018.8541485.

[163] A. J. Sanchez-Clemente, L. Entrena, and M. Garcia-Valderas, "Partial TMR in FPGAs Using Approximate Logic Circuits," *IEEE Transactions on Nuclear Science*, vol. 63, pp. 2233–2240, 4 Aug. 2016. DOI: 10.1109/TNS.2016.2541700.

[164] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," *Proceedings - 2013 18th IEEE European Test Symposium, ETS 2013*, 2013. DOI: 10.1109/ETS.2013.6569370.

[165] A. Bosio, I. O'Connor, G. S. Rodrigues, F. K. Lima, and S. Hamdioui, "Exploiting Approximate Computing for implementing Low Cost Fault Tolerance Mechanisms," *Proceedings - 2020 15th IEEE International Conference on Design and Technology of Integrated Systems in Nanoscale Era, DTIS 2020*, Apr. 2020. DOI: 10.1109/DTIS48698.2020.9081268.

[166] A. M. Keller and M. J. Wirthlin, "Partial TMR for Improving the Soft Error Reliability of SRAM-Based FPGA Designs," *IEEE Transactions on Nuclear Science*, vol. 68, pp. 1023–1031, 5 May 2021. DOI: 10.1109/TNS.2021.3070856.

[167] G. S. Rodrigues, J. S. Fonseca, F. L. Kastensmidt, V. Pouget, A. Bosio, and S. Hamdioui, "Approximate TMR based on successive approximation and loop perforation in microprocessors," *Microelectronics Reliability*, vol. 100-101, p. 113 385, Sep. 2019. DOI: 10.1016/J.MICROREL.2019.06.077.

[168] B. Pratt, M. Fuller, and M. Wirthlin, "Reduced-precision redundancy on FPGAs," *International Journal of Reconfigurable Computing*, vol. 2011, 2011. DOI: 10.1155/2011/897189.

[169] G. S. Rodrigues, J. Fonseca, F. Benevenuti, F. Kastensmidt, and A. Bosio, "Exploiting approximate computing for low-cost fault tolerant architectures," *Proceedings - 32nd Symposium on Integrated Circuits and Systems Design, SBCCI 2019*, Aug. 2019. DOI: 10.1145/3338852.3339875. [Online]. Available: https://doi.org/10.1145/3338852.3339875.

[170] "Reduced Precision Redundancy for Reliable Processing of Data," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, pp. 1960–1971, 4 2021. DOI: 10.1109/TETC.2019.2947617.

[171] X. Inc., *PG109 - Fast Fourier Transform v9.1 LogCORE IP Product Guide*, English, version 9.1, Xilinx Inc., 2021. [Online]. Available: https://docs.xilinx.com/r/en-US/pg109-xfft.

[172] A. Lindoso, M. Garcia-Valderas, L. Entrena, Y. Morilla, and P. Martin-Holgado, "Evaluation of the suitability of NEON SIMD microprocessor extensions under proton irradiation," *IEEE Transactions on Nuclear Science*, vol. 65, pp. 1835–1842, 8 Aug. 2018. DOI: 10.1109/TNS.2018.2823540.

[173] D. G. Mavis and P. H. Eaton, "Soft error rate mitigation techniques for modern microcircuits," *IEEE International Reliability Physics Symposium Proceedings*, vol. 2002-January, pp. 216–225, 2002. DOI: 10.1109/RELPHY.2002.996639.

[174] F. G. de Lima Kastensmidt, G. Neuberger, R. F. Hentschke, L. Carro, and R. Reis, "Designing fault-tolerant techniques for SRAM-based FPGAs," *IEEE Design and Test of Computers*, vol. 21, pp. 552–562, 6 Nov. 2004. DOI: 10.1109/MDT.2004.85.

[175] B. Pratt, M. Caffrey, J. F. Carroll, P. Graham, K. Morgan, and M. Wirthlin, "Fine-grain SEU mitigation for FPGAs using partial TMR," *IEEE Transactions on Nuclear Science*, vol. 55, pp. 2274–2280, 4 Aug. 2008. DOI: 10.1109/TNS.2008.2000852.

[176] M. Nicolaidis, *Soft errors in modern electronic systems*. 2010. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4419-6993-4.

[177] K. LaBel, M. S. .-. 2. N. E. Technology, and undefined 2016, "NEPP Roadmaps, COTS, and Small Missions," *ntrs.nasa.gov*, 2016. [Online]. Available: https://ntrs.nasa.gov/api/citations/20160007686/downloads/20160007686.pdf.

[178] M. Nicolaidis, "Design for soft error mitigation," *IEEE Transactions on Device and Materials Reliability*, vol. 5, pp. 405–418, 3 Sep. 2005. DOI: 10.1109/TDMR.2005.855790.

[179] A. Sanchez-Clemente, L. Entrena, M. Garcia-Valderas, and C. Lopez-Ongil, "Logic masking for SET Mitigation Using Approximate Logic Circuits," *Proceedings of the 2012 IEEE 18th International On-Line Testing Symposium, IOLTS 2012*, pp. 176–181, 2012. DOI: 10.1109/IOLTS.2012.6313868.

[180] A. Sanchez-Clemente, L. Entrena, and M. Garcia-Valderas, "Error masking with approximate logic circuits using dynamic probability estimations," *Proceedings of the 2014 IEEE 20th International On-Line Testing Symposium, IOLTS 2014*, pp. 134–139, 2014. DOI: 10.1109/IOLTS.2014.6873685.

[181] O. Ruano, J. A. Maestro, and P. Reviriego, "A methodology for automatic insertion of selective tmr in digital circuits affected by seus," *IEEE Transactions on Nuclear Science*, vol. 56, pp. 2091–2102, 4 Aug. 2009. DOI: 10.1109/TNS.2009.2014563.

[182] T. Jones. ""Xilinx Technologies for New Space/Space 2.0". Technical Education Webinar Series." (2021), [Online]. Available: https://event.on24.com/wcc/r/3607087/3822F579B2EFCEC03945380069DC0694 (visited on 03/16/2023).

[183] Microchip, *Radiation-Tolerant FPGAs. Space Solutions*, English, version DS00003023C, Microchip, 2020. [Online]. Available: https://www.microchip.com/en-us/products/fpgas-and-plds/radiation-tolerant-fpgas (visited on 12/09/2022).

[184] P. Reviriego, J. A. Maestro, I. López, and J. A. D. Agapito, "Soft error tolerant infinite impulse response filters using reduced precision replicas," *Proceedings of the European Conference on Radiation and its Effects on Components and Systems, RADECS*, pp. 493–496, 2011. DOI: 10.1109/RADECS.2011.6131424.

[185] K. Chen, L. Chen, P. Reviriego, and F. Lombardi, "Efficient implementations of reduced precision redundancy (RPR) Multiply and accumulate (MAC)," *IEEE Transactions on Computers*, vol. 68, pp. 784–790, 5 May 2019. DOI: 10.1109/TC.2018.2885044.

[186] W. Stechele, "Protecting fpga-based automotive systems against soft errors through reduced precision redundancy," *2015 10th IEEE International Symposium on Industrial Embedded Systems, SIES 2015 - Proceedings*, pp. 170–173, Aug. 2015. DOI: 10.1109/SIES.2015.7185057.

[187] A. Jung and P. E. Crouzet. "NIR HAWAII-2RG Data processing algorithms - Benchmarking software." (2018), [Online]. Available: https://essr.esa.int/project/nir-hawaii-2rg-data-processing-algorithms-benchmarking-software (visited on 05/24/2022).

[188] A. Lindoso, M. Garcia-Valderas, and L. Entrena, "Analysis of neutron sensitivity and data-flow error detection in arm microprocessors using neon simd extensions," *Microelectronics Reliability*, vol. 100-101, p. 113 346, Sep. 2019. DOI: 10.1016/J.MICROREL.2019.06.038.

[189] X. Inc., *PG149 - FIR Compiler v7.2 LogCORE IP Product Guide*, English, version 7.2, Xilinx Inc., 2022. [Online]. Available: https://www.xilinx.com/support/documents/ip_documentation/fir_compiler/v7_2/pg149-fir-compiler.pdf.

[190] X. Inc., *7 Series FPGAs Data Sheet: Overview*, English, version 2.6.1, Xilinx Inc., 2020, 20 pp. [Online]. Available: https://www.mouser.com/datasheet/2/903/ds180_7Series_Overview-1591537.pdf.

[191] D. Inc., *Nexys4 DDR FPGA Board Reference Manua*, English, version C, Digilent Inc., 2016, 26 pp. [Online]. Available: https://digilent.com/reference/_media/reference/programmable-logic/nexys-4-ddr/nexys4ddr_rm.pdf.

[192] D. Inc., *ZYBO FPGA Board Reference Manua*, English, version B, Digilent Inc., 2017, 26 pp. [Online]. Available: https://digilent.com/reference/_media/reference/programmable-logic/zybo/zybo_rm.pdf.

[193] D. Inc., *Zybo Z7 Board Reference Manua*, English, version B, Digilent Inc., 2018, 31 pp. [Online]. Available: https://digilent.com/reference/_media/reference/programmable-logic/zybo-z7/zybo-z7_rm.pdf.

[194] C. N. de Aceleradores. "Acelerador Ciclotrón 18/9 MeV." (2022), [Online]. Available: http://cna.us.es/index.php/es/instalaciones/ciclo (visited on 12/01/2023).

[195] P. S. Institut. "Welcome to the Proton Irradiation Facility." (2022), [Online]. Available: https://www.psi.ch/en/pif (visited on 12/01/2023).

[196] S. R. A. Laboratory. "ChipIR: instrument for the irradiation of Microelectronics." (2022), [Online]. Available: https://www.isis.stfc.ac.uk/Pages/Chipir.aspx (visited on 12/01/2023).

[197] R. P. Foundation, *Raspberry Pi 3 Model B+*, English, Raspberry Pi Foundation, 2018, 5 pp. [Online]. Available: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf.

A. MATERIALS AND METHODS

In this Chapter we will review the materials and methods we have employed to carry out the research for this Thesis. Although some of the materials and methods have already been presented in the pertaining Chapters, the goal of this Appendix is to provide more detailed information regarding the experimental setups utilized and how the data was processed to obtain the information presented in the articles.

A.1. Resources

For the experiments conducted, we used some support devices, facilities and systems. In this Section we will describe them to ensure repeatability of the results for anyone interested.

A.1.A. Devices Under Test

Throughout this work, we have used three models of Xilinx FPGAs and MPSoCs to develop and test the presented error mitigation techniques: Artix 7 [190], Zynq-7010 and Zynq-7020 [138]. These FPGAs were respectively mounted on Digilent Development Boards Nexys 4 [191], Zybo [192] and Zybo Z7 [193].

The majority of the experiments were carried out in the Zynq-7010 MPSoC because of its versatility and its proven hardness to latch-up, both in its Zybo and Zybo Z7 versions. The Zynq-7020 was used for experiments where the FPGA resources of the Zynq-7010 were not enough to accommodate the design we wanted to test. Some fault injection tests were also performed in the Artix 7 to test the consistency of the fault injection performed by Xilinx' Soft Error Mitigation IP.

The Zybo development boards were chosen because of the additional resources they offer, which are quite convenient for debugging and designing experiments. These boards comprise additional memory to be used by the processors, configuration through internal flash or external SD memories, and several connections that can be accessed from either the microprocessors or the programmable logic.

A.1.B. Radiation facilities

We conducted irradiation experiments in three facilities in the duration of the Thesis, although the results presented in this work stem from just two of them.

- **Centro Nacional de Aceleradores (CNA)**
  This facility, located in Sevilla (Spain), offers, among other radiation services, the

18 MeV proton accelerator we used for our experiments [194]. Although the energy of the particles present in this beam is not very high, the accelerator can provide high particle fluxes, up to $5 \times 10^8 p/cm^2 s$.

According to our experience, delidding the device is not necessary under such circumstances to provoke faults in the tested devices using a high particle flux.

- **Paul Scherrer Institut (PSI)**
  The PSI facility, located in Villigen (Switzerland), is a large laboratory where multiple science disciplines are studied [195]. Its Proton Irradiation Facility (PIF) provides a proton beam that can be adjusted at 5 energy stages from 230 to 74 MeV and degraded to achieve energies in between ranging from 230 to 6 MeV. According to their available information, the beam can achieve a maximum flux of $2 \times 10^9 p/cm^2 s$ at 230 MeV.

  The results from the experiments we performed in that facility were presented in [7].

- **ISIS Neutron and Muon Source**
  ISIS Neutron and Muon Source is a laboratory that produces neutron and muon beams to study different aspects of science. The ChipIR facility is at ISIS is a beam specially conceived to the irradiation of microelectronic components [196]. The neutron beam at ChipIR mimics the neutron atmospheric energy spectrum, with energies ranging from the thermal neutrons to the hundreds of MeVs, but with a 10^9 multiplying factor in its intensity. The flux of neutrons has been measured to be $5 \times 10^6 p/cm^2 s$ at energies higher than 10 MeV.

A.1.C. Experimental setup

In order to control and monitor the experiments, an external host computer has been used. For its versatility, ease to use and small size, a Raspberry Pi [197] device running a Linux distribution was selected.

The Raspberry Pi is equipped with an expansion card with electrical relays capable of independently controlling the power source of four Zybo Development boards, which allows easy and remote reconfiguration of the FPGA upon error.

The Raspberry has got 4 USB ports, that can be used to communicate via serial port with up to four devices under test. We used the USB ports to read information coming from the microprocessor serial port, a hardware implemented UART and the serial communication of the Soft Error Mitigation IP. Special cables to serve as interface from UART to the USB protocol are necessary in the last two cases.

The device also has an ethernet connection, which comes in handy to perform remote monitoring of the experiments running on the Zybos. For irradiation experiments, the

Raspberry Pi could be placed in the irradiation chamber and controlled from the bunker using just the ethernet connection.

Additionally, the GPIO pins of the Raspberry can be used to communicate and trigger certain actions in the Zybo. This was the strategy we followed to randomly start the fault injection through the Soft Error Mitigation IP. This way, we could make sure that the configuration memory addresses in which faults were injected were chosen randomly. A more detailed explanation about the injection mechanism is explained in the next Section.

Fig. A.1a shows the experimental setup used for the irradiation experiments and Fig. A.1b shows the modified setup for the fault injection experiments.

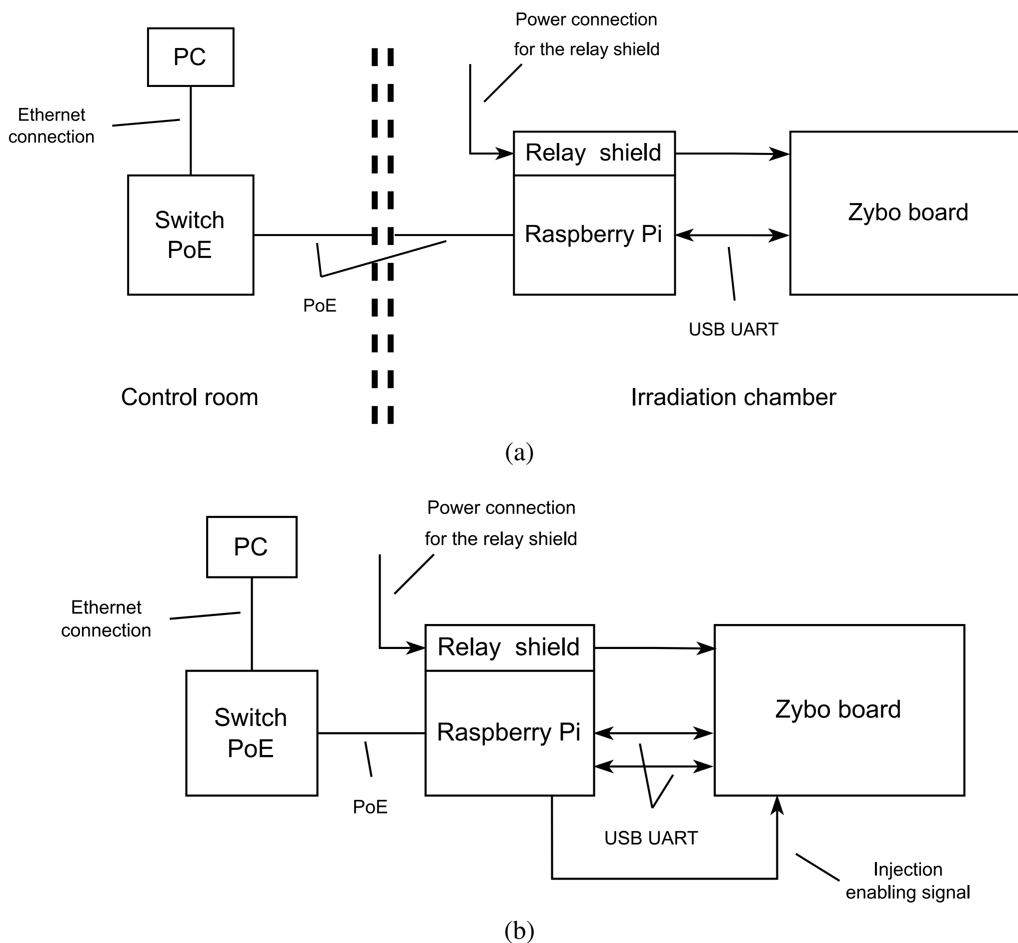


Figure A.1: Experimental setups for (a) the irradiation experiments and (b) the fault injection experiments.

Fig. A.2 displays the way this setup should be mounted either for fault injection or in the irradiation chamber.

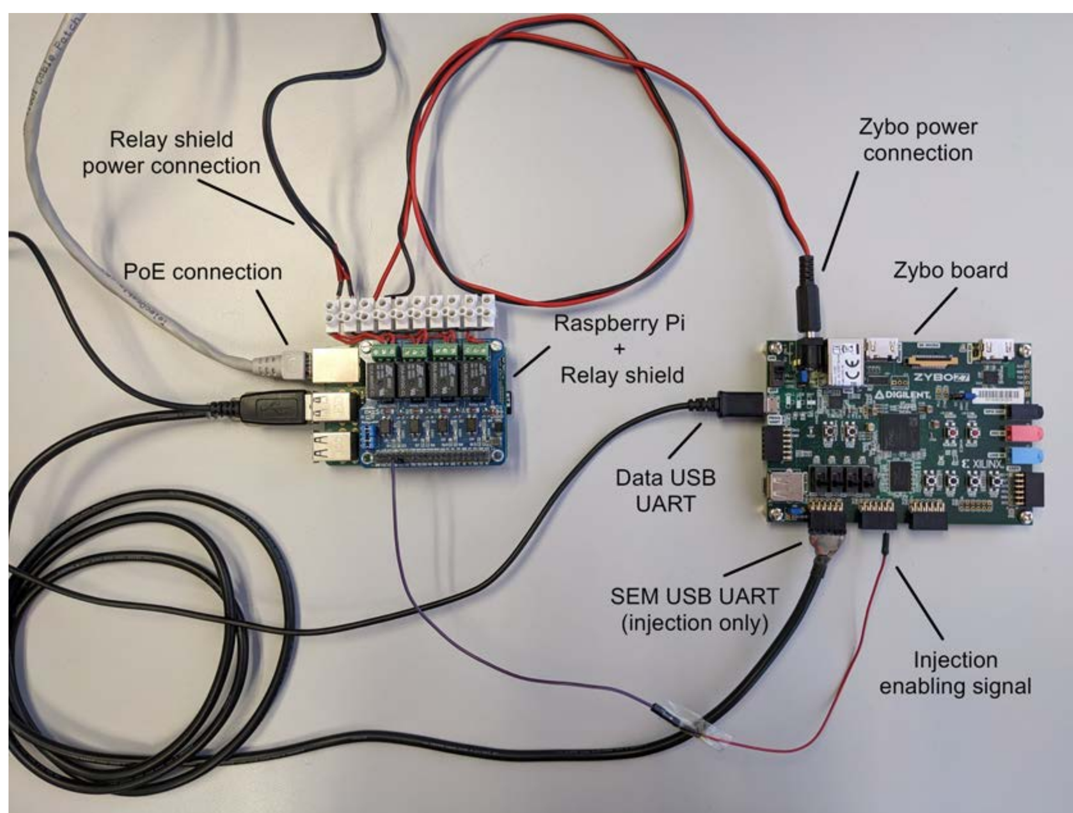


Figure A.2: Experimental setup.

A.2. Methodology

In this Section we will cover the more technical aspects of the methods used to perform the experiments, from the Fault Injection mechanism, to the acquisition of data using the previously reviewed setup and, finally, how the extracted data has been analyzed to obtain information.

A.2.A. Fault injection

Fault injection experiments have been carried out by implementing Xilinx Soft Error Mitigation (SEM) IP in the fabric of the FPGA. This IP is capable of performing scrubbing operations in the configuration memory of the FPGA and also inject faults in any specified bit of said memory.

The SEM IP can be controlled and observed via serial terminal and the fault injection can be effectively done using automation scripts for serial terminals. This allows for exhaustive fault injection in all the positions of the memory or random injection, by generating random memory addresses to inject using the controlling computer.

In this work, however, the generation of random memory addresses was performed in the FPGA itself. We instantiated a SEM controller module that provides true random

addresses to the IP. The controller is hardened using TMR to avoid errors caused by the fault injection.

The random number generator in the SEM controller needs a seed number to begin the calculations. This seed must change between power cycles to avoid the repetition of memory addresses generated by the controller. The seed generation relies on a counter that counts the number of clock cycles since the FPGA was configured and the seed value is selected as the number of that counter when an external signal is asserted. This signal can be activated manually (using a switch in the board) or automatically, using one of the GPIOs of the Raspberry Pi connected to a pin of the FPGA.

The SEM IP communicates through its UART interface some important information such as its state, the memory address it is injecting a fault on and information about the scrubbing process, i.e. the severity and position of the faults found and if the scrubber was able to correct it. By analysing the log output of the SEM IP in any fault injection campaign we could validate the randomness of the memory addresses in which the faults were injected.

A.2.B. Data acquisition

The data acquisition process is divided between the Zybo board and the Raspberry Pi.

A.2.B.1 Zybo

All the proposed and tested designs are instantiated in a testbench circuit responsible for providing inputs for the circuit under test and collecting its outputs. Additionally, this testbench contains error checking modules that compare the outputs of the (usually) three redundant modules in the circuit plus the voted output against the expected correct results. All the results are stored ina FIFO memory at the same time they are checked. A discrepancy between any of the four results and the expected results triggers the sending of the four packages of data to the Raspberry through a UART connection.

Some versions of the testbench implement a hardware UART in the FPGA, while others use a Direct Memory Access (DMA) controller so that the microprocessor reads the stored outputs in the Programmable Logic and sends them in the adequate format through one of its own UART ports.

All the resources in the testbench are triplicated and voted and the critical state machines have encoded states to reduce the sensitivity of the testbench against SEUs and SEFIs.

Fig. A.3 shows the schematic of the testbench employed in the experiments. The input/output generation and the error checkers are triplicated and voted in a TMR fashion to avoid errors in the data acquisition chain.

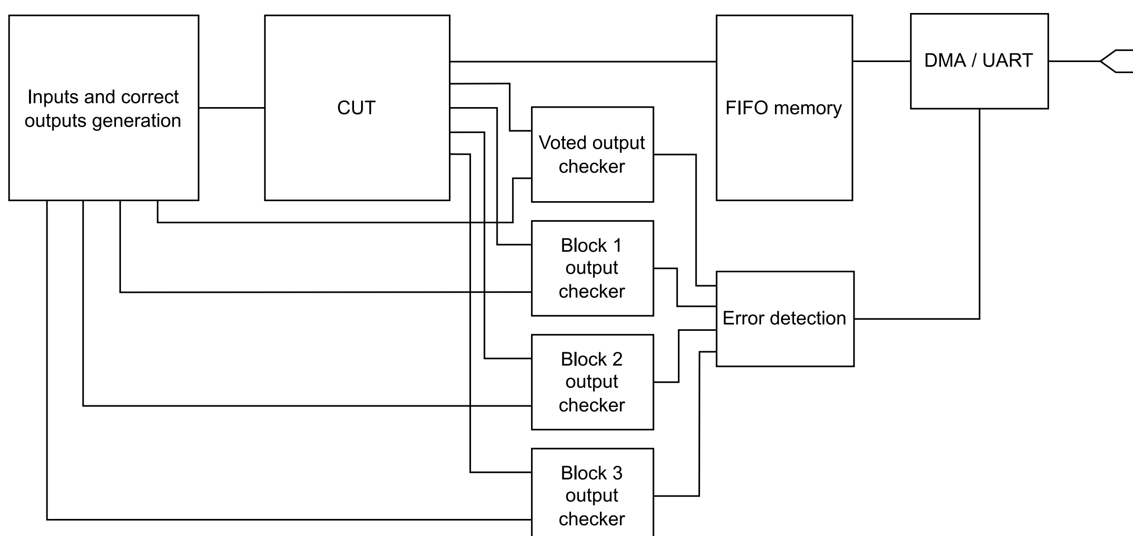


Figure A.3: Architecture of the testbench system.

This testbench can be easily replicated a number of times in order to fill the FPGA as much as possible and obtain all the results possible in a single irradiation campaign.

A.2.B.2 Raspberry Pi script

The packages containing the faulty data are logged by a script running on the Raspberry Pi. The script stores the data and adds a timestamp to each fault. With each fault, the script performs a power-cycle in the Zybo to reconfigure the FPGA and avoid error accumulation in the configuration memory.

The script running on the Raspberry also acts as an external watchdog to detect certain problems in the application implemented on the Zybo board. Loss of UART connection, long periods of time without faults or without any input from the Zybo, may denote some kind of functional interruption. This triggers a powercycling mechanism from the Raspberry to reconfigure and restart the FPGA.

A.2.C. Data processing

The logged data stored in the Raspberry Pi can be analyzed offline using automatic tools. For that purpose, the data processing has been divided in three steps, each carried out by a python script.

The first step is to extract the faults in the raw data log and also to extract and adequately format SEM log data, if present. This script also involves discarding corrupted characters that may occur due to faults in the UART transmission.

The second step is devoted to analyze the log produced by the SEM IP. The main goal of this step is to count the amount of addresses injected during the experiment

and to calculate the percentage of repeated addresses to evaluate the randomness of the injections.

The third and last step is to analyze the packages of data in which faults occurred. The data from the log is parsed and the results of the three redundant copies and the voted output is assigned to different arrays. For each data point, the correctness of the result is checked against the expected results and, in case of error, the error is classified according to the behaviour of the fault mitigation technique implemented in the design. The script carrying out this step must be carefully modified to emulate the voting logic of the mitigation technique and infer how did the fault affect the system and whether the voting was successful.

Additionally to the error classification, this last step also involves computing statistical analyses regarding the quality and modes of failures of the faults and generating graphical representations of the data, that helps in debugging and analyzing the error mitigation technique under consideration.