

# **A Self-adaptive Resilient Method for Implementing and Managing the High-reliability Processing System**

---

**Junchao Chen**

**Dissertation**

**zur Erlangung des akademischen Grades**

**Doktor der Ingenieurwissenschaften  
(Dr.-Ing.)**

**in der Wissenschaftsdisziplin "Rechnenarchitektur und Fehlertoleranz"**

**eingereicht an der  
Mathematisch-Naturwissenschaftlichen Fakultät  
Institut für Informatik  
der Universität Potsdam  
und  
IHP – Leibniz-Institut für innovative Mikroelektronik**

Potsdam, 03.02.2023.



Unless otherwise indicated, this work is licensed under a Creative Commons License Attribution 4.0 International.

This does not apply to quoted content and works based on other permissions.

To view a copy of this licence visit:

<https://creativecommons.org/licenses/by/4.0>

Hauptbetreuer: Prof. Dr.-Ing. Milos Krstic (*Universität Potsdam und IHP*)

Zweitbetreuer: Prof. Dr. Peter Langendörfer (*Brandenburgische Technische Universität und IHP*)

Gutachter 1: Prof. Dr.-Ing. Milos Krstic (*Universität Potsdam und IHP*)

Gutachter 2: Prof. Dr. Peter Langendörfer (*Brandenburgische Technische Universität und IHP*)

Gutachter 3: Prof. Dr.-Ing. Sergio Montenegro (*Universität Würzburg*)

Prüfungskommission:

1.Prof. Dr.-Ing. habil. Ulrike Lucked (*Universität Potsdam*) - *Vorsitzende*

2.Prof. Dr.-Ing. Milos Krstic (*Universität Potsdam und IHP*)

3.Prof. Dr. Peter Langendörfer (*Brandenburgische Technische Universität und IHP*)

4.Prof. Dr.-Ing. Sergio Montenegro (*Universität Würzburg*)

5.Prof. Dr. Michael Gössel (*Universität Potsdam*)

6.PD Dr. rer. nat. habil. Henning Bordihn (*Universität Potsdam*)

Published online on the

Publication Server of the University of Potsdam:

<https://doi.org/10.25932/publishup-58313>

<https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-583139>



# Acknowledgement

Throughout the writing of this dissertation, I have received a great deal of support and assistance. Therefore, I would like to express my gratitude to everyone who contributed to the successful completion of this work.

First and foremost, I want to express my gratitude to Prof. Milos Krstic, my supervisor, for his inexhaustible support and numerous guidance. I am privileged to have the opportunity to carry out this work in his research department at the IHP Microelectronics. Additionally, this endeavour would not have been possible without the generous support from the Maria Skłodowska-Curie RESCUE ETN project, which financed my research.

Then, I am pleased to acknowledge my colleagues and friends for their invaluable assistance in the past years. Special thanks to Marko Andjelkovic for illuminating my research and providing valuable support for my publications. For valuable assistance in my professional and personal life, I would like to thank Aleksandar Simevski, Anselm Breitenreiter, Dmytro Petryk, Klaus Tittelbach Helmrich, Li Lu, Markus Ulbricht, Navaneetha Channiganathota Manjappa, Oliver Schrape, Thomas Lange, Yuanqing Li. It is a joy to spend such a meaningful time with you all.

Last but not least, I would be remiss in not mentioning my family, especially my parents, Hongli Wang and Hong Chen, and my spouse, Ying Gao. Their belief in me has kept my spirits and motivation high during this challenging process.



# Abstract

As a result of CMOS scaling, radiation-induced Single-Event Effects (SEEs) in electronic circuits became a critical reliability issue for modern Integrated Circuits (ICs) operating under harsh radiation conditions. SEEs can be triggered in combinational or sequential logic by the impact of high-energy particles, leading to destructive or non-destructive faults, resulting in data corruption or even system failure. Usually, the SEE mitigation methods are deployed statically in processing architectures based on the worst-case radiation conditions, which is most of the time unnecessary and results in a resource overhead. Moreover, the space radiation conditions are dynamically changing, especially during Solar Particle Events (SPEs). The intensity of space radiation can differ over five orders of magnitude within a few hours or days, resulting in several orders of magnitude fault probability variation in ICs during SPEs. To overcome the static mitigation overhead issue, this thesis introduces a comprehensive approach for designing a self-adaptive fault resilient multiprocessing system. This work mainly addresses the following topics:

- Design of on-chip radiation particle monitor for real-time radiation environment detection
- Investigation of space environment predictor, as the support for solar condition forecast
- Dynamic mode configuration in the resilient multiprocessing system

Therefore, according to detected and predicted in-flight space radiation conditions, the target system can be configured to use no mitigation or low-overhead mitigation during non-critical periods. The redundant resources can be used to improve system performance or save power. On the other hand, during increased radiation activity periods, such as SPEs, the mitigation methods can be configured appropriately depending on the dynamic space environment, resulting in higher system reliability. Thus, a dynamic trade-off in the target system between reliability, performance, and power consumption in real-time can be achieved. All results of this work are evaluated in a highly reliable quad-core multiprocessing system that allows the self-adaptive setting of optimal radiation mitigation mechanisms during run-time. Proposed methods can serve as a basis for establishing a comprehensive self-adaptive resilient system design process. Successful implementation of the proposed design in a quad-core multiprocessor also shows its application perspective in other designs.

# Zusammenfassung

Infolge der CMOS-Skalierung wurden strahleninduzierte Einzelereignis-Effekte (SEEs) in elektronischen Schaltungen zu einem kritischen Zuverlässigkeitsproblem für moderne integrierte Schaltungen (ICs), die unter rauen Strahlungsbedingungen arbeiten. SEEs können in der kombinatorischen oder sequentiellen Logik durch den Aufprall hochenergetischer Teilchen ausgelöst werden, was zu destruktiven oder nicht-destruktiven Fehlern und damit zu Datenverfälschungen oder sogar Systemausfällen führt. Normalerweise werden die Methoden zur Abschwächung von SEEs statisch in Verarbeitungsarchitekturen auf der Grundlage der ungünstigsten Strahlungsbedingungen eingesetzt, was in den meisten Fällen unnötig ist und zu einem Ressourcen-Overhead führt. Darüber hinaus ändern sich die Strahlungsbedingungen im Weltraum dynamisch, insbesondere während Solar Particle Events (SPEs). Die Intensität der Weltraumstrahlung kann sich innerhalb weniger Stunden oder Tage um mehr als fünf Größenordnungen ändern, was zu einer Variation der Fehlerwahrscheinlichkeit in ICs während SPEs um mehrere Größenordnungen führt. In dieser Arbeit wird ein umfassender Ansatz für den Entwurf eines selbstanpassenden, fehlerresistenten Multiprozessorsystems vorgestellt, um das Problem des statischen Mitigation-Overheads zu überwinden. Diese Arbeit befasst sich hauptsächlich mit den folgenden Themen:

- Entwurf eines On-Chip-Strahlungsteilchen Monitors zur Echtzeit-Erkennung von Strahlung Umgebungen
- Untersuchung von Weltraumumgebungsprognosen zur Unterstützung der Vorhersage von solaren Teilchen Ereignissen
- Konfiguration des dynamischen Modus in einem belastbaren Multiprozessorsystem

Daher kann das Zielsystem je nach den erkannten und vorhergesagten Strahlungsbedingungen während des Fluges so konfiguriert werden, dass es während unkritischer Zeiträume keine oder nur eine geringe Strahlungsminderung vornimmt. Die redundanten Ressourcen können genutzt werden, um die Systemleistung zu verbessern oder Energie zu sparen. In Zeiten erhöhter Strahlungsaktivität, wie z. B. während SPEs, können die Abschwächungsmethoden dynamisch und in Abhängigkeit von der Echtzeit-Strahlungsumgebung im Weltraum konfiguriert werden, was zu einer höheren Systemzuverlässigkeit führt. Auf diese Weise kann im Zielsystem ein dynamischer Kompromiss zwischen Zuverlässigkeit, Leistung und Stromverbrauch in Echtzeit erreicht werden. Alle Ergebnisse dieser Arbeit wurden in einem hochzuverlässigen Quad-Core-Multiprozessorsystem evaluiert, das die selbstanpassende Einstellung optimaler



Strahlungsschutzmechanismen während der Laufzeit ermöglicht. Die vorgeschlagenen Methoden können als Grundlage für die Entwicklung eines umfassenden, selbstanpassenden und belastbaren Systementwurfsprozesses dienen. Die erfolgreiche Implementierung des vorgeschlagenen Entwurfs in einem Quad-Core-Multiprozessor zeigt, dass er auch für andere Entwürfe geeignet ist.

# Contents

<b>Acknowledgement</b> .....	V
<b>Abstract</b> .....	VII
<b>Zusammenfassung</b> .....	VIII
<b>List of Figures</b> .....	XIII
<b>List of Tables</b> .....	XVIII
<b>List of Acronyms</b> .....	XX
<b>1. Introduction</b> .....	1
1.1 Motivation .....	2
1.2 Research Scope and Objectives.....	4
1.3 Thesis Contribution.....	5
1.4 Thesis Organization.....	6
1.5 Publications .....	7
<b>2. Fundamentals</b> .....	9
2.1 Space Radiation Environment.....	9
2.1.1 Solar Particle Events .....	12
2.2 Radiation Effects in Integrated Circuits.....	15
2.2.1 Single-Event Effects .....	16
2.3 Dependability of Systems.....	19
2.3.1 Threats to dependability.....	19
2.3.2 Means to achieve dependability.....	20
2.3.3 Attributes of dependability .....	22
2.4 Supervised Machine Learning.....	23
2.5 Summary .....	27
<b>3. State of the Art</b> .....	28
3.1 Particle Detector.....	28
3.1.1 Bulk Build-in Current Detectors.....	29
3.1.2 Acoustic Wave Detectors.....	30
3.1.3 Diode Detectors .....	31
3.1.4 3D NAND Flash Detector.....	32
3.1.5 Pulse Stretching Inverter Detector .....	33

3.1.6	RAM-based Detectors.....	34
3.1.7	Comparison of Particle Detectors.....	35
3.2	Space Weather and System Reliability Analysis.....	36
3.2.1	Space Weather Recognition.....	37
3.2.2	System Reliability Analysis.....	38
3.3	Dynamic Adaptation Systems.....	40
3.3.1	FPGA-based Dynamic Systems.....	40
3.3.2	ASIC-based Dynamic Systems.....	42
3.4	Summary of Open Issues in this Thesis.....	44
<b>4.</b>	<b>SRAM-based Radiation Monitor.....</b>	<b>46</b>
4.1	Concept.....	46
4.2	Single-Event Upset Monitor Design.....	53
4.2.1	Scrubbing.....	54
4.2.2	EDAC.....	55
4.2.3	Detection flow.....	57
4.2.4	Fault Management Block.....	60
4.3	Implementation of the SEU Monitor.....	60
4.3.1	External SRAM-based implementation.....	61
4.3.2	Internal SRAM-based implementation.....	62
4.4	Analysis of Results.....	66
4.4.1	SEU Monitor Performance Analysis.....	66
4.4.2	SEU Sensitivity Analysis.....	67
4.4.3	Synthesis Results.....	69
4.5	Summary.....	70
<b>5.</b>	<b>Solar Particle Event Prediction.....</b>	<b>72</b>
5.1	Concept.....	72
5.2	Historical Solar Condition Analysis.....	75
5.2.1	Historical Flux Data Collection.....	76
5.2.2	SPE Energy Spectra Reconstruction.....	81
5.2.3	SEU Rate Estimation.....	86
5.3	SEU Prediction with Machine Learning.....	91
5.3.1	Pre-processing of the Dataset.....	92
5.3.2	Model Training.....	93
5.3.3	Model Evaluation.....	95
5.3.4	Online Parameter Adjustment.....	99

5.4	Hardware Accelerator Implementation .....	102
5.4.1	Single Event Upset Prediction .....	104
5.4.2	Online Parameter Adjustment.....	105
5.5	Analysis of Results.....	106
5.5.1	Prediction Performance Analysis.....	106
5.5.2	Online Learning Performance Analysis.....	109
5.5.3	Synthesis Results .....	113
5.6	Summary .....	113
<b>6.</b>	<b>Self-Adaptive Fault Resilience System.....</b>	<b>115</b>
6.1	Concept.....	115
6.2	Reconfigurable Resilience Methods.....	118
6.2.1	Core-level NMR.....	118
6.2.2	DVFS .....	126
6.3	Optimal Mode Selection.....	129
6.3.1	Mode selection for core-level NMR .....	130
6.3.2	Mode selection on DVFS.....	134
6.3.3	Concurrent selection of core-level NMR and DVFS.....	137
6.4	Analysis of Results.....	140
6.5	Summary .....	144
<b>7.</b>	<b>Conclusion .....</b>	<b>146</b>
7.1	Achieved Results .....	147
7.2	Future Work .....	148
	<b>Bibliography .....</b>	<b>150</b>

# List of Figures

Figure 1. 1: General process of the self-adaptive resilient system design process. The purple blocks represent the design phase methods, the green blocks represent the running phase methods, and the blue dash blocks represent the main focus of this dissertation.....	3
Figure 2. 1: Categorization of the space radiation natural sources. The green blocks represent the main focus of this thesis. ....	10
Figure 2. 2: Peak directional proton flux (5-minute averages for energies > 10 MeV) and the number of solar events which affect the Earth's environment each year according to [NOAA]. Moreover, the number of solar cycles is marked.....	12
Figure 2. 3: Particle flux of an idealized Solar Particle Event (SPE) with a possible Energetic Storm Particle (ESP) over time. ....	13
Figure 2. 4: Tree structure of IC reliability failure mechanism and corresponding fault classification. The green block represents the main focus of the dissertation. ....	15
Figure 2. 5: Simplified ionization mechanisms responsible for radiation-induced faults: direct (left) and indirect (right) ionization. ....	17
Figure 2. 6: Dependability-related terms are arranged in a tree. The green blocks represent the main focus of this dissertation.....	19
Figure 2. 7: Failure propagation sequence.....	19
Figure 2. 8: Bathtub curve representing failure rate [ASI14]. ....	22
Figure 3. 1: Example of a PMOS-BBICS implementation for SET detection. It detects particle strikes at PMOS transistors [GWF07]. ....	30
Figure 3. 2: Typical cantilever sensing structure for acoustic wave detector [UVG16]...	31
Figure 3. 3: Example of a processing channel for a diode-based detector [ACS <sup>+</sup> 20]. ....	32
Figure 3. 4: Example architecture of a 3-D NAND floating gate array [MBA <sup>+</sup> 20]. ....	32
Figure 3. 5: Basic structure of two-inverter Pulse-Stretching Cell (PSC) [ACS <sup>+</sup> 21]. ....	33
Figure 3. 6: Simplified architecture of the RAM array. Hollow squares represent storage cells, which are various for different RAM types.....	34
Figure 4. 1: General concept of the proposed monitor with the sensing elements inside an embedded system. The green blocks represent the main focus of this dissertation. ....	47
Figure 4. 2: Architecture of a six transistor SRAM cell. BT and BC are the two-bit lines connected to the SRAM cell through the two access transistors (Mpg1 and Mpg2). The two nodes (QC and QT) store the information [CAS <sup>+</sup> 19]. ....	49

Figure 4. 3: General workflow of the proposed monitor with four main modules. ....	54
Figure 4. 4: Sensing elements working mode decision process.....	57
Figure 4. 5: Dedicated faults detection flowchart of the proposed SEU monitor with the HSAIO SEC-DED EDAC code, scrubbing and ODRF.....	58
Figure 4. 6: Workflow of the Over-counting Detection Register File (ODRF).....	59
Figure 4. 7: Block diagram of the fault management unit. ....	60
Figure 4. 8: Master-slave architecture of the 20-Mbit MCM SSRAM with SEU monitor. ....	61
Figure 4. 9: Mesh of trees 4*8 communication architecture for a quad-core system with eight internal SRAM banks. The proposed monitor is integrated into the light blue parts.....	63
Figure 4. 10: Misaligned data access procedure in the memory interface controller. ....	64
Figure 4. 11: Block diagram of the standard current injection approach in SPICE simulation to evaluate target element SET/SEU critical charge. ....	67
Figure 4. 12: Example of the SRAM cell's critical charge in supply voltage during current injection experiment in SPICE simulation. ....	68
Figure 5. 1: General concept of the SEU monitor with solar condition predictor inside a space-borne system. The green blocks represent the main focus of this dissertation. ....	73
Figure 5. 2: General flow of the proposed solar condition prediction method. The green blocks represent the main focus of this chapter.....	74
Figure 5. 3: Evaluation process of the historical solar condition analysis model for collecting the SEU rate dataset of the target SRAM-based SEU monitor from past solar events. The green blocks represent the main focus of this chapter. ....	75
Figure 5. 4: Example of proton flux during January 17-31, 2012, from GOES-13. Channels 1~7 from the EPEAD detector and 8~11 from the HEPAD detector. The dashed line (the proton with energy10 MeV) respects the event line.....	79
Figure 5. 5: Example of heavy-ion flux during January 17-31, 2012 from ACE-SIS. The detection channel of each ion is also present in each corresponding sub-figures. Some data gaps are encountered for a few ions. The channel 1~7 from the EPEAD detector and the channel 8~11 from the HEPAD detector. The dash line (proton with energy10 MeV) respects the event line. ....	81
Figure 5. 6: Weibull fit of the target 65 nm SRAM heavy-ion and proton static bit cross-section results from radiation tests [VGU17]. ....	82
Figure 5. 7: Example of ACE-SIS hourly carbon ion flux energy spectra reconstruction for extrapolation to higher energies on 02 March 2012, 00:00. Dashed lines correspond to the corresponding extrapolation method flux fitting results. Squares correspond to the ACE-SIS hourly carbon flux. The coloured lines are the existing carbon flux data with different CRÈME 96 SPE models. WW	

stands for the worst week, WD means worst day, and P5M stands for peak five minutes. ....	83
Figure 5. 8: Example of GOES hourly proton ion flux energy spectra reconstruction on 02 March 2012, 00:00. Dashed lines correspond to the corresponding extrapolation method flux fitting results. Triangles correspond to the hourly proton in the GOES low energy detector. Circles correspond to the hourly proton in the GOES HEPAD detector. WW stands for the worst week, WD means worst day, and P5M stands for peak five minutes.....	85
Figure 5. 9: CREME96 flow chart for calculating heavy-ion and proton-induced upset rates. A dashed block indicates an optional calculation step. The green blocks represent the main focus of this chapter. ....	86
Figure 5. 10: Example of unshielded and shielded (100 mils of aluminium) directional proton flux in differential form as a function of the energy for the five existing CRÈME 96 space environment modules.....	87
Figure 5. 11: Example of unshielded and shielded (100 mils of aluminium) directional heavy-ion flux in the integral form as a function of the LET for the five existing CRÈME 96 space environment modules.....	88
Figure 5. 12: March 6-11, 2012 SRAM hourly SEU rate estimated from GOES proton database. The particle flux for all lower to higher energy channels is shown, and all channel data are of good quality.....	90
Figure 5. 13: March 6-11, 2012, SRAM hourly SEU rate estimated from ACE-SIS heavy-ion database. The particle ion flux of He, C and Ni for the lower and higher energy channels is shown, and the data is of poor quality. ....	90
Figure 5. 14: Evaluation process of the model training process for collecting, selecting and training machine learning models for space environment prediction. .	91
Figure 5. 15: $R^2$ scores (higher the better) for the selected regression models with varying history data length nh.....	96
Figure 5. 16: $RMSE$ scores (lower the better) for the selected regression models with varying history data length nh.....	96
Figure 5. 17: Block diagram of the recurrent neural network with long short term memory layer.....	98
Figure 5. 18: Block diagram of the proposed online SRAM SEU rate prediction function parameter adjustment procedure. ....	100
Figure 5. 19: $R^2$ scores for the online parameter adjustment evaluation on the dataset. ....	101
Figure 5. 20: Evaluation process of the proposed hardware accelerator model for performing the real-time space environment prediction and online prediction adjustment. ....	102
Figure 5. 21: Proposed hardware accelerator design with the interface to external logic. ....	103
Figure 5. 22: Decision flowchart of the proposed hardware accelerator. ....	103

Figure 5. 23: Jan 19-31, 2012, hardware accelerator SEU prediction performance for 2Gbit and 4 Mbit SRAM during one large SPEs, respectively. ....	107
Figure 5. 24: Mar 07-15, 2011, hardware accelerator SEU prediction performance for 2Gbit and 4 Mbit SRAM during one small SPEs, respectively. ....	108
Figure 5. 25: Mar 07-15, 2011, hardware accelerator SEU prediction performance for 20 Mbit during one small SPEs.....	109
Figure 5. 26: The gradient descent with contour plot with respect to the <i>Coef1</i> and <i>Coef3</i> under the different online learning rates.....	110
Figure 5. 27: Online learning prediction on a middle-size solar event performs from scratch (i.e. no offline dataset available). The initial coefficients of the prediction equation (i.e. trained 0 instances) are set to all 0. ....	111
Figure 5. 28: The prediction performance for online learning optimizes the offline prediction function on the new data set (i.e. with different shielding performance). ....	112
Figure 6. 1: General concept of Self-Adaptive Fault Resilience system. The green blocks represent the main objectives of this dissertation. ....	116
Figure 6. 2: Decision tree for determining the operation mode. The grey blocks are the main focus of this chapter. ....	117
Figure 6. 3: Example of possible core-level NMR operating mode in a quad-core multiprocessing system. The circle corresponds to the core. Four circles in the solid or dashed rectangle correspond to the quad-core system. The hollow circle represents the deactivated core with power-saving mechanisms. Different colours represent different tasks running on the cores.....	119
Figure 6. 4: General architecture of a SAFR quad-core processing system. The core-level NMR modes are implemented and controlled in the framework controller. The green blocks represent the main focus of this dissertation. ....	120
Figure 6. 5: General framework controller block diagram. Thick and thin lines denote data and control flow, respectively. CG stands for clock gating. APB bridge is the part for peripheral interconnect control of the multiprocessing system [AS114].. ....	121
Figure 6. 6: Example of OML, IML configuration in the framework controller. Core 0,1 and 2 form the core-level TMR, and core 3 is in the high-performance mode [AS114].. ....	122
Figure 6. 7: Example of user-defined TMR in a quad-core multiprocessing system. Different colours represent different tasks running on the cores.....	123
Figure 6. 8: Example of self-TMR in a quad-core multiprocessing system. Different colours represent different tasks running on the cores. The hollow square represents the clocked- or powered-gating of the core.....	123
Figure 6. 9: Comparison of reliability functions of core-level NMR in a quad-core system. The default means the de-stress or high-performance mode. The core	



failure rate $\lambda_c = 1$ is assumed, and the QMR mode is shown as an ideal condition which $Pd = 1$ . .....	125
Figure 6. 10: Decision flow of the reconfigurable system optimal mode selection. ....	129
Figure 6. 11: Example of SAFR optimal mode selection for the core-level NMR method in a quad-core system. SEUT is in-flight hourly SEUs from the sensor network; SEU0 is average hourly SEUs during background conditions. The reference normalized value is obtained from the high-performance or de-stress mode of the quad-core system. ....	133
Figure 6. 12: Example of system performance and power consumption with respect to the SEUT/SEU0 in the SAFR optimal mode selection of the quad-core system. The reference value for performance and power consumption corresponds to a value from single-core. ....	133
Figure 6. 13: Decision flow of the proposed DVFS adjustment strategy to increase system reliability. $PFHth$ is the pre-defined DVFS trigger threshold with SIL standard. ( $fs$ , $vs$ ) are the system frequency and voltage. ....	134
Figure 6. 14: Example of SAFR optimal mode selection for the DVFS method in a system. SEUT is the in-flight hourly SEUs from the sensor network; SEU0 is the average hourly SEUs during background conditions. The default operation can be the high-performance or de-stress operation mode. ....	136
Figure 6. 15: Example of system performance and power consumption with respect to SEUT/SEU0 with the DVFS method. The baseline of performance and power consumption is the corresponding behaviour of ( $fmax$ , $vmin$ ) with a single-core. ....	136
Figure 6. 16: Decision flow of the core-level NMR and DVFS methods to increase system reliability in the proposed SAFR system. ....	137
Figure 6. 17: Example of SAFR optimal mode selection for a quad-core system's core-level NMR and DVFS methods. SEUT is the in-flight SEUs from the sensor network; SEU0 is the average SEUs during background conditions. The default value is obtained from the cores in high-performance or de-stress operation mode with ( $fmax$ , $vmin$ ). ....	139
Figure 6. 18: Example of system performance and power consumption with respect to the SEUT/SEU0 in the SAFR optimal mode selection of the quad-core system with core-level NMR and DVFS method. The baseline of performance and power consumption is the corresponding behaviour of a single core. ....	140
Figure 6. 19: Example of SAFR system optimal mode selection with core-level NMR and DVFS during large and small solar events, respectively. The system's default operation is the high-performance or de-stress mode. ....	142

# List of Tables

Table 2. 1: Measured on-board SRAM upset rate (upsets/bit/day) in a geostationary satellite during large SPEs [DLH <sup>+</sup> 07].....	14
Table 3. 1: Comparison of radiation particle detectors.....	36
Table 4. 1: The suitability of ASIC on-chip components as sensing elements. The SRAM is chosen as the sensing element in this thesis. .....	48
Table 4. 2: Weibull fitting parameters for the target SRAM heavy-ion bit or event cross-section with static or dynamic operating mode. ....	52
Table 4. 3: Weibull fitting parameters for the target SRAM proton bit or event cross-section with static or dynamic operating mode. ....	52
Table 4. 4: Heavy-ion induced SEU rate with the static and dynamic Weibull fitting parameters in different CREME96 space radiation particle models. ....	53
Table 4. 5: Proton induced SEU rate with the static and dynamic Weibull fitting parameters in different CREME96 space radiation particle models .....	53
Table 4. 6: Fe ion (LET = 19 Mev.cm <sup>2</sup> /mg) radiation test results of a 16 Mbit SRAM with and without ECC for static and dynamic tests [VGU17]. ....	56
Table 4. 7: Critical charge for SRAM and different standard cells. ....	69
Table 4. 8: Area comparison (in $\mu\text{m}^2$ ) between radiation-hardened controllers with and without proposed SEU monitor. ....	70
Table 4. 9: Power consumption comparison (in mW) between radiation-hardened controllers with and without proposed SEU monitor. ....	70
Table 5. 1: Selected solar event lists from NOAA during solar cycle 24. Proton fluxes are integral 5-minute averages for energies > 10 MeV, given in Particle Flux Units (pfu), measured by GOES spacecraft at Geosynchronous orbit: 1 pfu = 1 p/sq. cm-s-sr. The measured X-rays are used to classify the importance of SPEs.....	77
Table 5. 2: Target SRAM SEU rate (upsets/bit/day) sensitivity to ions with different existing CRÈME 96 SPE models. ....	82
Table 5. 3: Example of the pre-processed SEU rate dataset around the start point (highlight in green) of one solar event on 07 March 2012. "Upset rate" is in unit SEU/bit/day, "Upset" is in unit SEU/2Gbit/hour the normalized upset is the min-max scaling of the hourly upset data; <i>nh</i> last hours (i.e. three last hours in this example, highlighted in grey) are the normalized upsets rate collected from previous hourly upset rates.....	92

Table 5. 4: Area and power comparison of the proposed space environment hardware accelerator with a 20 Mbit SRAM and the proposed SEU monitor. ....	113
Table 6. 1: Probability of Failure per Hour (PFH) for different Safety Integrity Levels (SILs) as defined in the IEC 61508 standard [IEC05].....	131
Table 6. 2: Simulated Non-SPE period SEU monitor network behaviour of the target 2 Gbit SRAM during solar cycle 24. ....	140
Table 6. 3: Simulated SPE period SEU monitor network behaviour of the target 2 Gbit SRAM during solar cycle 24. SEU0 is the baseline SEU rate, which is set to $2.80 * 10^{-7}$ upsets/bits/day. ....	141
Table 6. 4: Example of the optimal mode selection of the SAFR system with core-level NMR and DVFS methods in a quad-core multiprocessor. SEU0 is the baseline SEU rate of the target SRAM, which is estimated as $2.80 * 10^{-7}$ upsets/bits/day.....	142
Table 6. 5: Example calculation of the system power consumption and performance as well as system reliability (i.e. time of satisfying SIL 1 requirement) comparison of different modes for a quad-core multiprocessor in one year. SAFR system1 and system2 refer to the default operation mode during the non-SPE period with de-stress and high-performance mode, respectively.	143

# List of Acronyms

<b>ACE</b>	Advanced Composition Explorer
<b>ANN</b>	Artificial Neural Network
<b>AMBA</b>	Advanced Microcontroller Bus Architecture
<b>ASIC</b>	Application Specific Integrated Circuit
<b>AVF</b>	Architectural Vulnerability Factor
<b>BBICS</b>	Bulk Built-in Current Sensor
<b>CME</b>	Coronal Mass Ejections
<b>CMOS</b>	Complementary Metal-Oxide Semiconductor
<b>CRÈME</b>	Cosmic Ray Effects on Micro-electronics Model
<b>DAEC</b>	Double Adjacent Error Correction
<b>DED</b>	Double Error Detection
<b>DFVS</b>	Dynamic Voltage Frequency Scaling
<b>DPU</b>	Data Processing Unit
<b>DRAM</b>	Dynamic Random Access Memory
<b>DSCOVR</b>	Deep Space Climate Observatory
<b>DSP</b>	Digital Signal Processor
<b>ECC</b>	Error Correction Code
<b>EDAC</b>	Error Detection And Correction
<b>EPEAD</b>	Energetic Proton, Electron, and Alpha Detector
<b>ESP</b>	Energetic Storm Particle
<b>FEC</b>	Forward Error Correction
<b>FPGA</b>	Field Programmable Gate Array
<b>FU</b>	Functional Unit
<b>GCR</b>	Galactic Cosmic Ray
<b>GEO</b>	Geosynchronous Equatorial Orbit
<b>GGS</b>	Global Geospace Science
<b>GOES</b>	Geostationary Operational Environmental Satellite
<b>GPU</b>	Graphics Processing Unit

<b>HEPAD</b>	High Energy Proton and Alpha Detector
<b>HUP</b>	Direct Ionization-Induced SEE Rate Calculation
<b>IC</b>	Integrated Circuit
<b>IML</b>	Input Multiplexing Logic
<b>IVF</b>	Instruction Vulnerability Factor
<b>LET</b>	Linear Energy Transfer
<b>LSTM</b>	Long Short-Term Memory
<b>LSU</b>	Load Store Unit
<b>LUT</b>	Look-Up Table
<b>MAE</b>	Mean Absolute Error
<b>MAX</b>	Maximum Absolute Error
<b>MBU</b>	Multiple Bit Upset
<b>MCM</b>	Multi-Chip Module
<b>MCU</b>	Multiple Cell Upset
<b>ML</b>	Machine Learning
<b>MLP</b>	Multi-Layer Perceptron
<b>MOSFET</b>	Metal-Oxide-Semiconductor Field-Effect Transistor
<b>MoT</b>	Mesh-of-Tree
<b>MSE</b>	Mean Squared Error
<b>MTTU</b>	Mean Time To Upset
<b>NMR</b>	N-Modular Redundant
<b>NOAA</b>	National Oceanic and Atmospheric Administration
<b>ODRF</b>	Over-counting Detection Register File
<b>OML</b>	Output Multiplexing Logic
<b>PDF</b>	Probability Density Function
<b>PFH</b>	Probability of Failures per Hour
<b>PFU</b>	Particle Flux Unit
<b>PSC</b>	Pulse-Stretching Cell
<b>PUP</b>	Proton-Induced SEE Rate Calculation
<b>PVF</b>	Program Vulnerability Factor
<b>P5M</b>	Peak Five Minutes
<b>RAM</b>	Random Access Memory
<b>RMSE</b>	Root Mean Square Error

<b>RNN</b>	Recurrent Neural Network
<b>RPP</b>	Rectangular Parallelepiped Parallelogram
<b>RVF</b>	Register Vulnerability Factor
<b>SAA</b>	South Atlantic Anomaly
<b>SAFR</b>	Self-Adaptive Fault Resilience
<b>SBU</b>	Single Bit Upset
<b>SDR</b>	Software Defined Radio
<b>SEB</b>	Single Event Burnout
<b>SEDR</b>	Single-Event Dielectric Rupture
<b>SEE</b>	Single Event Effect
<b>SEFI</b>	Single-Event Functional Interrupt
<b>SEGR</b>	Single Event Gate Rupture
<b>SEL</b>	Single Event Latch-up
<b>SEM</b>	Space Environment Monitor
<b>SER</b>	Soft Error Rate
<b>SET</b>	Single Event Transient
<b>SEU</b>	Single Event Upset
<b>SGD</b>	Stochastic Gradient Descent
<b>SIL</b>	Safety Integrity Level
<b>SIS</b>	Solar Isotope Spectrometer
<b>SMU</b>	Single-word Multiple-bit Upset
<b>SOHO</b>	Solar and Heliospheric Observatory
<b>SOI</b>	Silicon On Insulator
<b>SoC</b>	System on Chip
<b>SPE</b>	Solar Particle Event
<b>SRAM</b>	Static Random Access Memory
<b>SSRAM</b>	Synchronous Static Random Access Memory
<b>SV</b>	Sensitive Volume
<b>SVM</b>	Support Vector Machine
<b>SWPC</b>	Space Weather Prediction Center
<b>TAEC</b>	Triple Adjacent Error Correction
<b>TCMI</b>	Tightly Coupled Memory Interconnect
<b>TID</b>	Total Ionizing Dose

<b>TMR</b>	Triple Modular Redundancy
<b>VLIW</b>	Very Long Instruction Word
<b>WD</b>	Worst Day
<b>WW</b>	Worst Week
<b>YFRR</b>	Youngest-First Round-Robin





# Chapter 1

## Introduction

Space electronic systems are being developed at an enormous speed nowadays, and considerably impact many applications, such as satellite communications, Earth observation, situational awareness, science & technology, etc. Many countries and companies are involved in research and development in the aerospace sector, with several companies popping up every quarter in launch vehicles, satellites, propulsion, and manufacturing. As a result, the total number of orbiting satellites has proliferated in the past few years, with 4,852 active satellites in orbit by the end of 2021, an increase of about 43.9% over last year [UCS22]. Moreover, the total revenue in 2021 in terms of global space revenue, government procurement, and government organization support is estimated to be USD 370 billion, with the potential to generate revenue of more than USD 1 trillion or more by 2040 [ESS22]. Due to the complexity of the space environment and the characteristics of space-borne applications, such as long missions and difficult maintenance, aerospace applications have a tremendous demand for high-reliability electronic systems, especially for radiation-hardened products. Furthermore, as semiconductor technology advances and nanoscale Integrated Circuits (ICs) become more complicated, electronic systems confront rising reliability threats, making nanoscale ICs more vulnerable to various faults, especially radiation-induced faults in aerospace [EPE11]. Therefore, in addition to standard metrics (i.e., area, power consumption, and performance), IC designers for space-borne applications must also consider reliability as a primary design metric.

The aim of this dissertation is to advance the current reliability design methodology by developing a self-adaptive resilient system for processing units, particularly multiprocessors in space applications. This chapter goes over general concepts. Section 1.1 presents the motivation for this work. Section 1.2 introduces the research scope and goals of this thesis. The main research contributions are detailed in Section 1.3. Section 1.4 presents the organization of this thesis. At the end of this chapter, the related publications for this thesis are listed in Section 1.5.

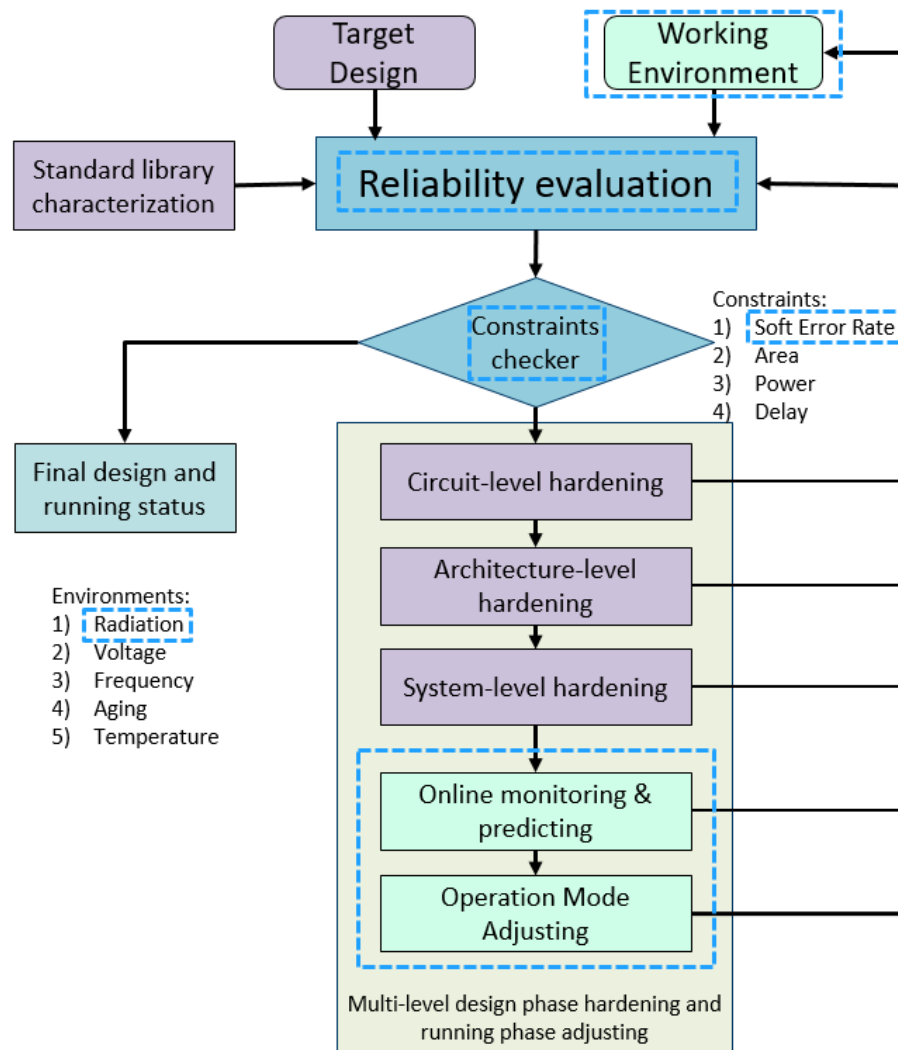
## 1.1 Motivation

Modern Complementary Metal-Oxide Semiconductor (CMOS) technology, which is currently primarily used to implement Application Specific Integrated Circuits (ASICs), is facing increasing reliability issues as transistor sizes decrease and supply voltage is aggressively reduced [MAN22]. As a result, current integrated circuits are vulnerable to a wide range of faults, caused by physical mechanisms inherent in semiconductor technology or environmental conditions, such as manufacturing problems, process variations, aging, electrostatic discharge, crosstalk-induced noise, electric and electromagnetic interference, and radiation-induced effects. Among these fault sources, radiation-induced effects from high-energy particles are significant reliability issues for space-born electronic systems. Radiation-induced effects in this thesis mainly consider Single Event Effects (SEEs), which primarily manifest as data corrupted by high-energy particles and may induce temporary or permanent faults in electronic systems. Thus, effective radiation-induced faults mitigation in electronic systems for space-born applications is essential.

High-energy particles in space come from various natural resources, and the corresponding ion fluxes change dynamically [EPE11]. One of the primary radiation sources is Solar Particle Events (SPEs), which can dominate the radiation environment when it occurs, resulting in a high flux of energetic particles. During SPEs, the intensity of cosmic radiation can vary by five orders of magnitude in a matter of hours or days, thus, significantly increasing the probability of radiation-induced effects in electronic spacecraft [EPE11]. High-energy particles, such as protons and heavy ions, can cause SEEs on electronic systems through direct or indirect ionization, potentially leading to malfunction or system failure [DWM03]. Data from previous space missions [RHS<sup>+</sup>01] [KHY<sup>+</sup>14] suggest that during peak SPE periods, SEE rates for space applications can increase hundreds to thousands of times.

Different mitigation schemes can be used to mitigate radiation-induced effects at various system layers (i.e., circuit, logic, architecture, software, and algorithm layers) [ECH<sup>+</sup>16]. The traditional method of dealing with system failures is to use static redundancy methods in hardware, time, software, and/or information at separate system layers [JCK18]. These methods, however, could induce significant overhead, which is unacceptable for many applications. Moreover, high-reliability designs typically statically apply mitigations based on worst-case scenarios, to achieve in each moment corresponding to worst-case system requirements, which is most of the time unnecessary and results in a resource overhead [GRH16]. As a result, in order to overcome the static mitigation overhead issue in a dynamic environment, mitigation mechanisms must be dynamically activated, in which deployed mitigation methods can be dynamically optimized according to real-time radiation environment conditions. In order to achieve the above target, the corresponding research question can be asked: how to detect and predict the real-time radiation environment, and how to perform the dynamic mitigation methods selection to maintain the optimal trade-off between system reliability, performance, and power consumption in real-time?

The proposed approach in this thesis is based on utilizing a self-adaptive resilient system, which can automatically select optimal operating modes according to detected and predicted space radiation conditions as well as application requirements for space-borne applications. Therefore, during relaxation with respect to space radiation, the system can be configured to use no mitigation or low-overhead mitigation. Thus, the released resources can be used to improve system performance or save power. Conversely, during radiative enhancements, such as SPEs, the mitigation methods can be configured appropriately with minimal overhead depending on the dynamic environment. An example is the adaptive multiprocessing system, which can dynamically adapt the fault-tolerant modes (e.g., core-level triple module redundancy) or low-protection modes (e.g., high performance and low power consumption) as the radiation conditions change.



**Figure 1.1:** General process of the self-adaptive resilient system design process. The purple blocks represent the design phase methods, the green blocks represent the running phase methods, and the blue dash blocks represent the main focus of this dissertation.

## 1.2 Research Scope and Objectives

This work is conducted in the framework of the Marie Skłodowska-Curie RESCUE project [MSC17], which has been focused on developing innovative methodologies for the interconnected challenges of reliability, security, and quality in electronic system design. The primary target of this research is to look into the self-adaptive use of resilient mechanisms in multiprocessing architectures, which are the foundation of modern embedded systems. Aside from that, the goal is also to investigate the dynamic trade-off between system reliability, performance, and power consumption in mixed-criticality systems.

Figure 1.1 presents the general process of designing a self-adaptive resilient system. This process mainly combines two development stages: design-phase system static hardening and running-phase monitoring & dynamic system adjusting. This method comprises a trade-off between system reliability as specified by application needs and design restrictions such as chip space, processing speed, power consumption, and dynamic working environment.

During the design phase (represented by the purple blocks in Figure 1.1), the developer needs to characterize and model the target system's reliability/area/power/delay requirements, as well as the target working environment case condition (such as the worst-case condition). Accordingly, the necessary essential hardening methods can be applied. Using a custom-designed rad-hard library is one of the most straightforward hardening approaches. However, the cost of this methodology is generally substantial, and it would also significantly increase system area and power consumption. Therefore, a common approach uses a non-radiation hardening standard library that provides acceptable performance with appropriate hardening design measures. In order to design a robust digital system during the design phase, there are a set of existing resilience techniques at various abstraction system design layers (i.e., circuit, architecture, system layers, etc.) [ECH<sup>+</sup>16]. In general, solutions for improving system reliability, particularly fault-tolerance systems, include some form of redundancy. Fault-tolerant mechanisms are broadly classified into four types based on the kind of redundancy they introduce, namely hardware, time, software, and/or information. Aside from that, complex systems may employ or combine a variety of fault-tolerance mechanisms. Thus, the design requirements (reliability, area, power, etc.) could be met by a suitable combination of these existing techniques across multiple abstraction layers.

System monitoring and dynamic adjustment during the running phase (represented by the green blocks in Figure 1.1) are critical for a self-adaptive system dealing with dynamic variations in working conditions, such as dynamic radiation environment, variant task requirements, system aging, temperature, etc. Moreover, for the designs intended to be applied in harsh radiation environments, designers commonly deploy design-phase static mitigation techniques based on the assumed worst-case condition. Therefore, the run-time system adjustment methods can be applied to an adaptive system, which could achieve the trade-off between two or more parameters (such as reliability, aging, power, performance, etc.). Online parameter and environment monitoring are crucial for self-adaptive systems, which require various on-chip monitors, such as radiation

monitor [ACS<sup>+</sup>20], aging monitor [SKK11], temperature sensor [SSB<sup>+</sup>20], etc. Moreover, it is also essential to predict the variations in the working environment to enable the implementation of necessary mitigation measures as early as possible. After obtaining the in-flight system and working environment status, the target system can accordingly adjust the operating mode based on the pre-defined reliability evaluation methods, thus, proposing an optimized efficiency approach and offering an optimal run-time solution.

This thesis mainly addresses the running-phase blocks in the introduced self-adaptive resilient system design process in Figure 1.1. Three main aspects are addressed in this thesis: 1) particle detector design for onboard in-flight radiation environment monitoring, which is critical for the application in the complex radiation environment, such as space; 2) radiation environment and the system reliability behavior detection and prediction method, which is necessary to provide the early protection of the reliability-critical system; 3) optimal online mode dynamic selection method by considering the design-phase reconfigurable mitigation approaches with the run-time system reliability evaluation. As a result, a self-adaptive fault resilient system for harsh radiation environments, such as space, can be obtained, which is capable of real-time radiation particle detection, in-flight working environment prediction, and self-adaptive optimal operation mode selection. The proposed method is designed for highly reliable and adaptive processing systems used in space applications, and it allows for trade-offs between system reliability, power consumption, and performance.

### 1.3 Thesis Contribution

Several methods for the implementation of a self-adaptive resilient system in radiation-harsh environments have been proposed in order to address the presented research objectives. The following are the primary contributions:

- SRAM-based Single Event Upset (SEU) Monitor: a non-standalone embedded SRAM-based radiation monitor that combines the SEU monitoring and data storage functions in the same on-chip Static Random Access Memory (SRAM) module, with negligible cost and overhead compared to traditional stand-alone SEU monitors. The internal or external SRAM blocks on a system can be applied as sensing elements. With a dedicated detection flow, the proposed monitor can enable the correction or detection of single and multiple-bit upsets as well as permanent faults in each memory word.
- Solar particle event and single event upset rate prediction: this work introduces an embedded approach for the prediction of SPEs by combining the proposed radiation monitor with the supervised machine learning model and space ion flux databases obtained from previous space missions. The proposed predictor can utilize the in-flight real-time SEU rate data and a pre-trained prediction model for forecasting the following SEU rate, thus, determining the corresponding solar radiation conditions. Therefore, upcoming flux changes and corresponding SPEs can be predicted at least one hour in advance, and the

target system can drive the appropriate radiation hardening mechanisms before the burst of high-energy particles.

- Implementation and evaluation of a self-adaptive system operation: the Self-Adaptive Fault Resilience (SAFR) system, an autonomous reconfigurable fault mitigation methodology for reliability-critical applications, is presented, analyzed, and implemented in a quad-core multiprocessing system. The SAFR system's primary function is to ensure system reliability in a complex and dynamic radiation environment, achieving an optimized trade-off between system reliability, power consumption, and performance. The SAFR system combines the SEU monitor, SPE & SEU rate predictor, and several dynamic mechanisms for multiprocessing systems. In dynamic radiation environments, the proposed SAFR system can use the detected and predicted radiation environment data, as well as the pre-defined reliability requirement, to determine the optimal operating mode in real-time.

## 1.4 Thesis Organization

The remainder of the thesis is divided into six chapters, which are as follows:

Chapter 2 elaborates on the fundamental aspects of the proposed self-adaptive resilient system for space-borne applications. The space radiation environment, especially solar events, is introduced and analyzed. Subsequently, the radiation effects of integrated circuits, especially SEEs, are discussed. This chapter also introduces the basic concepts of dependable system design, as well as corresponding evaluation and mitigation methods. In addition, the concept of supervised machine learning methods is introduced.

Chapter 3 provides an in-depth overview of related cutting-edge research, including particle detectors for online radiation monitoring, space weather and system reliability prediction and evaluation, and dynamic adaptive systems. The most critical and relevant existing achievements are summarized for each of these topics, and the open issues addressed in this thesis are outlined.

The design of the proposed on-chip SRAM-based SEU monitor is presented in Chapter 4. First, the general concept of monitors is introduced. The architecture and operation of the proposed SRAM-based SEU monitor are described. Based on the proposed method, two specific implementation cases for the monitor are introduced, one based on external stand-alone SRAM and the other based on internal SRAM blocks for multiprocessing systems. In addition, the proposed monitor design is analyzed accordingly, and the radiation sensitivity of SRAM is analyzed in detail. The chapter concludes with a summary of the achieved results.

Chapter 5 is devoted to predicting the SPE and SEU from supervised machine learning and the on-chip SEU monitor. The concept of the prediction method is presented first. The analyzing method for the historical solar events during solar cycle 24 (2008-2019) is detailed. As a result, the space radiation particle-induced SEU rate dataset for the target SRAM during historical solar events can be obtained. Using the collected dataset with multiple time-series regression machine learning models, a supervised learning model can be selected for the real-time SEU and space

environment prediction. Moreover, the online learning method is also introduced for prediction optimization during run-time. Besides that, a dedicated low-cost hardware accelerator design is detailed for implementing selected regression models and online algorithms. The prediction model's performance under various solar events is shown for the analysis of the results. The optimization performance of the online learning method and the implementation consumption of the dedicated hardware accelerator are also detailed. The chapter concludes with a summary of the results obtained.

Chapter 6 introduces the proposed SAFR system for multiprocessing systems. Based on the proposed radiation monitor and predictor, the general concept of the SAFR with cross-layer dynamic methods is presented at the beginning. As a research example, the use of two common dynamic methods (i.e., core-level n-module redundancy and dynamic voltage and frequency scaling) in a quad-core system is detailed. Subsequently, according to detected and predicted radiation environments as well as system reliability requirements, the optimal system operating mode selection methodology is introduced. The proposed methodology is then evaluated using a quad-core multiprocessing system and historical solar events. The chapter is concluded with an outline of the achieved results.

Chapter 7 summarizes the most significant accomplishments and elaborates on the primary strategic directions for future work.

## 1.5 Publications

Most of the results presented in this thesis have been published in peer-reviewed journals and at national and international conferences, as follows:

### Granted Patent:

- [JC1] J. Chen, M. Krstic, M. Andjelkovic, A. Simevski, "Electronic Circuit With Integrated SEU Monitor", German patent, 16/893.753, 25.10.2022.

### Journal publications:

- [JC2] J. Chen, T. Lange, M. Andjelkovic, A. Simevski, and M. Krstic, "Prediction of Solar Particle Events with SRAM-Based Soft Error Rate Monitor and Supervised Machine Learning," *Microelectronics Reliability*, Volume 114, 2020, doi: 10.1016/j.microrel.2020.113799.
- [JC3] J. Chen, T. Lange, M. Andjelkovic, A. Simevski, L. Lu and M. Krstic, "Solar Particle Event and Single Event Upset Prediction from SRAM-based Monitor and Supervised Machine Learning," in *IEEE Transactions on Emerging Topics in Computing (TETC)*, 2022, doi: 10.1109/TETC.2022.3147376.

**Conference publications:**

- [JC4] J. Chen, M. Krstic, "Engineering of Cross-Layer Fault Tolerance in Multiprocessing Systems" Proc. PhD Forum of the 8th BELAS Summer School, Tallinn, June, 2018, Estonia.
- [JC5] J. Chen, M. Andjelkovic, A. Simevski, Y. Li, P. Skoncej, and M. Krstic, "Design of SRAM-Based Low-Cost SEU Monitor for Self-Adaptive Multiprocessing System," In Proc. 22nd Euromicro Conference on Digital System Design (DSD), Kalithea, Greece, 2019, doi: 10.1109/DSD.2019.00080.
- [JC6] J. Chen, M. Krstic, "Self-Adaptive Single-Event Upsets Mitigation for Dependable Multiprocessing Systems" Proc. PhD Forum of the 9th BELAS Summer School, Frankfurt-Oder, June, 2019, Germany.
- [JC7] J. Chen, T. Lange, M. Andjelkovic, A. Simevski, and M. Krstic, "Prediction of Solar Particle Events with SRAM-Based Soft Error Rate Monitor and Supervised Machine Learning," in Proc. 31st European Symposium on Reliability of Electron Devices, Failure Physics and Analysis (ESREF), online, 2020.
- [JC8] J. Chen, T. Lange, M. Andjelkovic, A. Simevski, and M. Krstic, "Hardware Accelerator Design with Supervised Machine Learning for Solar Particle Event Prediction," 2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Frascati, Italy, 2020, doi: 10.1109/DFT50435.2020.9250856.

Furthermore, the author of this thesis has co-authored 3 peer-reviewed journal articles and 6 conference papers, all of which are partly related to the topics investigated in this work. These publications are listed as references in the Bibliography: [AVC<sup>+</sup>19] [ACS<sup>+</sup>20] [MAN<sup>+</sup>20] [LBA<sup>+</sup>20] [ACS<sup>+</sup>21] [LCB<sup>+</sup>21] [RSA<sup>+</sup>21] [MKR<sup>+</sup>21] [ASC<sup>+</sup>22].



## Chapter 2

# Fundamentals

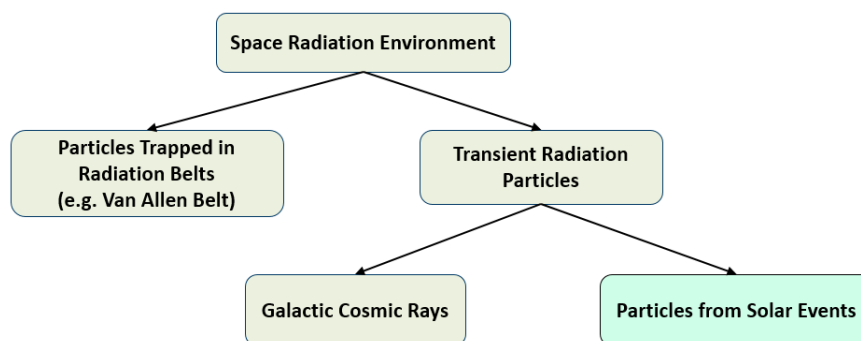
The rising prevalence of IC defects necessitates the development of novel solutions to improve system dependability in a variety of ways. The principles for constructing a self-adaptive resilient system for space applications are presented in this chapter. First, the space radiation environment is presented and evaluated, particularly the Solar Particle Event (SPE), which is the crucial environmental effect for space-borne application design, as discussed in Section 2.1. Subsequently, the radiation-induced effects of integrated circuits, especially Single Event Effects (SEEs), are discussed in Section 2.2. Understanding various aspects of SEEs and the corresponding effects on circuits is essential for analyzing and designing dependable systems. Basic concepts for the design of dependable systems, as well as evaluation methods and mitigation approaches, are mentioned in Section 2.3. The supervised machine learning method is introduced in Section 2.4. The machine learning methods are important for self-adaptive systems to perceive and predict the working environment, and then make corresponding adjustments. The chapter is concluded in Section 2.5.

### 2.1 Space Radiation Environment

The impact of the space radiation environment on the design and adjustment of space-borne applications is critical. On Earth, the atmosphere and magnetic field shield humans from the majority of space radiation. When leaving the Earth's natural shelter, however, different forms of radiation can be experienced. According to [BMX08], the space radiation environment is responsible for around 20% of satellite abnormalities. Therefore, understanding and analyzing the space radiation environment is essential to addressing design and application challenges and achieving reliable, cost-effective space applications.

Radiation is defined as the emission or transmission of energy in the form of waves or particles through space or a material medium, and it is typically classified as ionizing or non-ionizing based on the energy of the emitting particles [WEW14]. In space, the main concern is ionizing radiation, which carries more than 10 eV and consists of gamma rays and particle radiation, such as electrons, positrons, neutrons, protons, and heavy-ion. Moreover, in nuclear physics, a heavy-ion refers to

any particle with one or more unit charges and a mass exceeding the mass of a proton. Some publications exclude the alpha particle (i.e., the helium-4 nucleus) from the heavy-ion. However, in this thesis, according to [CRÈ96], the analysis of radiation-induced effects by alpha particles on electronic circuits is also included in the standard analysis procedure of heavy-ion induced effects. As shown in Figure 2.1, the natural source of space radiation particles can be divided into two categories [BDS03]: protons, electrons, and heavy-ions trapped in "belts" by planetary magnetospheres; and transient radiation particles from deep space, such as galactic ray, which includes protons and heavy-ions of all elements in the periodic table. These phenomena will be detailed in the following paragraphs.



**Figure 2.1:** Categorization of the space radiation natural sources. The green blocks represent the main focus of this thesis.

The theory of radiation trapping in planetary magnetospheres was first proposed by Birkeland in 1895 when he conducted a vacuum chamber experiment to study auroras [BDS03]. Subsequently, scientists discovered that the planet's magnetic field can trap charged particles, creating radiation belts. The Van Allen belt is the primary source of trapped charged particles on Earth, mostly protons (the inner belt) and electrons (the inner and outer belt). Furthermore, heavy-ions can also be trapped in planetary magnetic fields, but they are less plentiful and, as a result, are unable to pierce spacecraft shielding materials. The highest concentration of protons, i.e., the proton band, is in the orbit of  $L_{Earth} = 1.7$ , where  $L_{Earth}$  expresses the distance in relation to the Earth's radius, i.e., approximately 6371 km. The flux in the proton band is very stable, with maximum energies ranging from a few MeV to hundreds of MeV. The situation for the electron belt is more complex, since there are two maxima for inner and outer belts, respectively: the inner belt is centered on  $L_{Earth} = 1.4$  and extends up to  $L_{Earth} = 2.8$ , and has a stable electron flux with a maximum energy of around 10 MeV; the outer belt is centered on  $L_{Earth} = 5$  and extends from  $L_{Earth} = 2.8$  to  $L_{Earth} = 10$ , with a variable electron population and energy levels up to 7 MeV. Furthermore, because the Earth's dipole is tilted and off-centered, these belts are all shifted by about 500 km towards the West Pacific and are inclined by roughly  $11^\circ$  to the rotating axis. As a result, the inner Van Allen radiation belt is closest to the Earth's surface in this region, known as

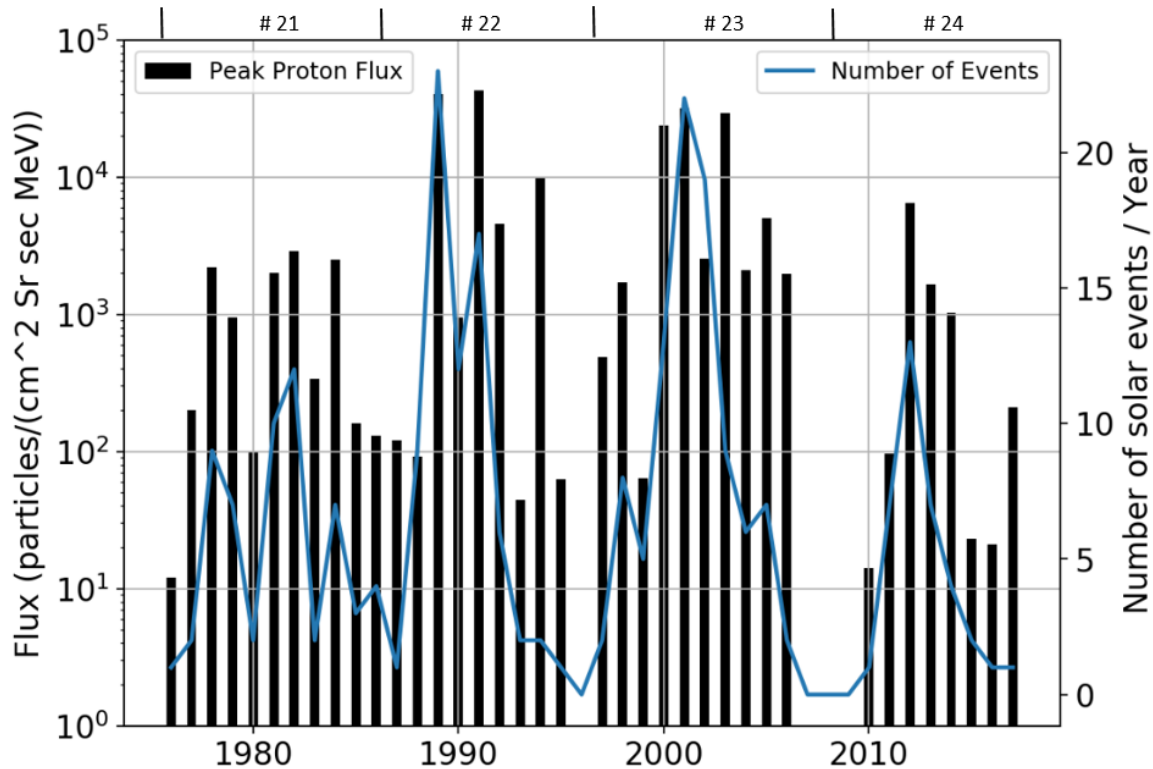
the South Atlantic Anomaly (SAA). Furthermore, an increase in the flux of energetic particles in the SAA region would expose orbiting satellites to more radiation than usual [BMX08]. Besides that, spacecrafts passing through low-Earth orbit enter the radiation region of the Van Allen radiation belt, and satellites in the Geostationary Earth Orbit (GEO) are out of the influence from this belt.

Transient radiation is made up of Galactic Cosmic Ray (GCR) particles and particles from solar phenomena such as coronal mass ejections and flares. These two space radiation phenomena are natural sources of galactic rays. Before going into detail about these transient radiation phenomena, it's important to highlight the solar activity cycle, which is an important part of modelling the space radiation environment and modifying transient radiation phenomena [BMX08]. A solar cycle is described as the reversal of the Sun's extremely active magnetic field every 11 years. Solar Cycle #24 is the most recent full solar cycle (2008-2019). In each active solar cycle, there is typically a 7-year period of high activity level, i.e., solar maximum with large numbers of sunspots appearing, and a 4-year period of low activity level, i.e., solar minimum with a low activity of sunspots. Although the shift between the solar maximum and minimum phases is uninterrupted, it is typically seen as abrupt for convenience. The Sun's magnetic polarity affects only the GCRs flux in space radiation conditions, not the trapped particle or SPEs flux [BMX08]. As a result, the cosmic radiation environment is often studied in 11-year cycles.

GCRs are slowly changing high-energy charged particles that originate outside the solar system, such as from the explosion of a supernova [BDS03]. GCRs include all of the natural elements in the periodic table (up to uranium) and are mostly composed of 85% protons, 14% helium, and 1% heavy-ions. Particles from GCR can be of high energy, up to  $10^{11}$  GeV, with a low flux, which is generally a few  $\text{cm}^{-2}\text{s}^{-1}$ , and could also vary with the solar activity. Moreover, the magnetic field in the Sun and solar wind could attenuate the GCR flux greatly during the solar maximum period, and a majority cannot penetrate Earth. In contrast, during the solar minimum period, it is easier for GCR particles to enter Earth in the absence of many solar events and their corresponding magnetic fields [TDB97]. These GCR particles are only a modest hazard to people and systems on the ground, but they may readily pass through or remain in satellite systems, depositing enough energy to cause failures in space-born systems.

There are several forms of infrequent and irregular eruptions on the Sun. The two most important are solar flares and Coronal Mass Ejections (CMEs), both of which involve substantial energy explosions but are otherwise distinct [MGL14]. The CME is a large emission of plasma and magnetic fields from the solar corona into the solar wind. CMEs release a plasma mostly consisting of electrons and protons, which can create shocks that drive high-energy, long-duration SPEs [AJT97]. Solar flares are massive bursts of electromagnetic radiation from the Sun, mostly electron-rich, that can last from minutes to hours. Sudden bursts of electromagnetic energy travel at the speed of light, so any impact on Earth's sunlight-exposed atmosphere would coincide with the observed event. In general, these two phenomena sometimes occur at the same time, especially

the strongest flares are almost always associated with CMEs. But they also have many differences, such as emitting different particles, travelling differently, and could have different effects around planets [MGL14]. Furthermore, in contrast to CMEs, solar flares typically do not generate large fluxes and energetic particles. This thesis mainly focuses on the impact of the SPE phenomenon on the space radiation environment, which is detailed in the following subsection.



**Figure 2.2:** Peak directional proton flux (5-minute averages for energies > 10 MeV) and the number of solar events which affect the Earth's environment each year according to [NOAA]. Moreover, the number of solar cycles is marked.

### 2.1.1 Solar Particle Event

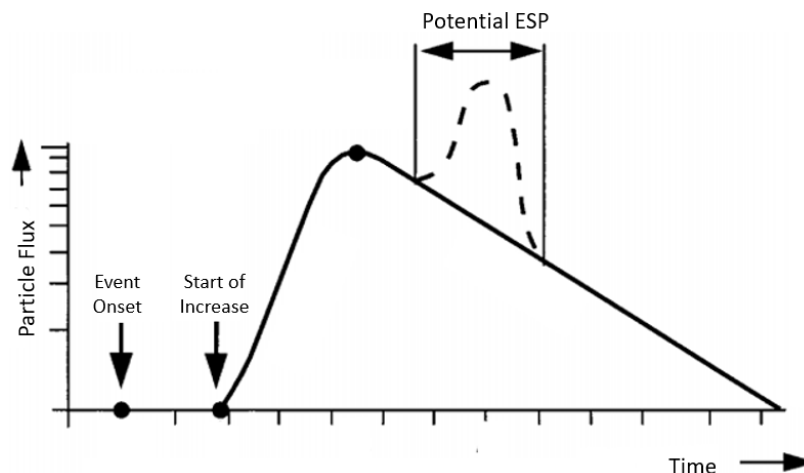
SPEs are induced by the Sun when the occupation of a large-scale magnetic eruption is generally driven by shocks of fast CMEs and associated solar flares. SPEs can dominate the radiation environment during its occurrence, resulting in a high flux of energetic particles, mostly protons. Other particles, such as heavy-ions, can also be accelerated during solar events. A solar flare can accelerate these particles in the solar atmosphere, or a CME can also accelerate them in interplanetary space. These high-energy particles are significant radiation hazards to space-born applications and astronauts. SPEs can often attain a peak flux in minutes or hours and then progressively drop over hours or days. The peak flux of SPEs is two to five orders of magnitude more than the amount of background space radiation [BDS03]. In addition, SPE events not only

have a great impact on the space radiation environment, but also have an impact on the terrestrial radiation environment even with the protection of the magnetic field and atmosphere. According to [BDS03], ground-based cosmic rays could increase levels by up to 5000% during large SPEs.

When an SPE happens, it has the capacity to become a major contributor to space radiation particles. According to the space environment center of the National Oceanic and Atmospheric Administration (NOAA) [NOAA], the start of one SPE is at least three consecutive five-minute intervals of flux data points  $\geq 10 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$  of the proton with energy  $\geq 10 \text{ MeV}$ . This means that at least ten particles of sufficient energy pass through a square centimeter of surface per second and per steradian to meet the SPE starting criteria. In addition, the end of the solar event is defined as the last time the flux  $\geq 10 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$ . NOAA has recorded a total of 267 SPE events affecting the Earth's environment since 1976 [NOAA]. SPEs generally occur between 0 to 30 times per year, and some are powerful enough to damage space-born electronic systems. Figure 2.2 presents all measured directional proton flux during SPEs since 1976 from NOAA. Moreover, the solar cycle intensity for the occurrence SPEs have been scaled by NOAA into five levels:

- S1 minor:  $\geq 10^1 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$  with 50 occurrences per solar active cycle
- S2 moderate:  $\geq 10^2 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$  with 25 occurrences per solar active cycle
- S3 strong:  $\geq 10^3 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$  with 10 occurrences per solar active cycle
- S4 severe:  $\geq 10^4 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$  with 3 occurrences per solar active cycle
- S5 extreme:  $\geq 10^5 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$  with 1 occurrences per solar active cycle

Besides that, the NOAA's Space Weather Prediction Center (SWPC) offers a variety of space weather forecasting services, particularly forecasts for the probability of S1 occurrence and warnings for expected S1 or higher events.



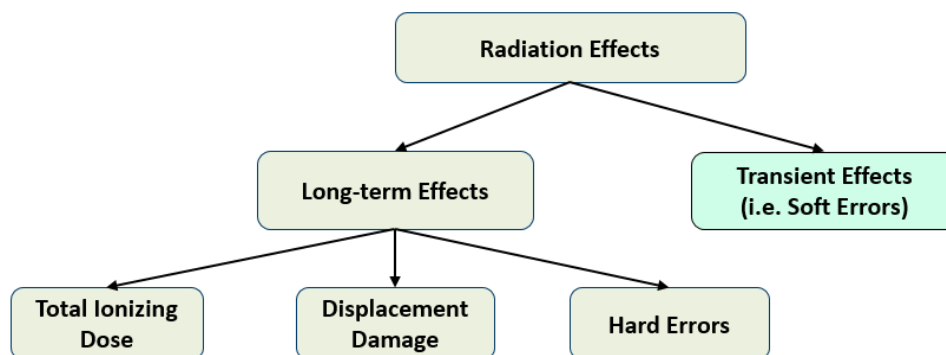
**Figure 2.3:** Particle flux of an idealized Solar Particle Event (SPE) with a possible Energetic Storm Particle (ESP) over time.

Figure 2.3 shows an ideal SPE with an exponentially fast-rising slope and a reciprocal exponentially slow-falling slope. According to [RTU00], one SPE event can be described using six parameters, namely, start time, peak flux time, interval to rise to maximum flux, the magnitude of maximum flux, duration, and end time. While each SPE event would be very different, they typically have a propagation delay of 20 to 90 minutes between initial acceleration and onset of particle augmentation; initially anisotropic flux, but gradually becomes more isotropic; tens of minutes to hours to reach peak flux rapidly; slowly exponential decrease to background radiation levels over a few hours to days. Furthermore, the SPE might involve a short period of flux augmentation, i.e., the Energetic Storm Particle (ESP), which can result in an increase in peak particle flux. During the ESP, many particles are trapped in the propagating shock front through wave-particle interactions, forming an ion pre-shock, so fast, and powerful interplanetary shock transfer can be observed. In general, ESP can lead to rapid increases in particle flux by up to two orders of magnitude [RTU00].

**Table 2.1:** Measured on-board SRAM upset rate (upsets/bit/day) in a geostationary satellite during large SPEs [DLH<sup>+</sup>07].

Date	Background	Worst Five Minutes	Worst Day	Worst Weeks
<b>April 15, 2001</b>	$3.7 \cdot 10^{-8}$	$3.8 \cdot 10^{-5}$	$6.1 \cdot 10^{-7}$	$1.3 \cdot 10^{-7}$
<b>Nov. 5, 2001</b>	$3.8 \cdot 10^{-8}$	$2.5 \cdot 10^{-5}$	$7.4 \cdot 10^{-7}$	$2.1 \cdot 10^{-7}$
<b>Oct. 28, 2003</b>	$4.4 \cdot 10^{-8}$	$2.5 \cdot 10^{-5}$	$6.1 \cdot 10^{-7}$	$2.1 \cdot 10^{-7}$
<b>Jan. 20, 2005</b>	$8.1 \cdot 10^{-8}$	$2.4 \cdot 10^{-5}$	$6.5 \cdot 10^{-7}$	$2.3 \cdot 10^{-7}$

SPE can have several severe effects in space but also on the terrestrial level. When high-energy particles hit satellites or humans in orbit, they can pierce the item and cause hazards to electronic circuitry or biological DNA. Moreover, high-flying aircraft, as well as passengers, may also be exposed to radiation risks, creating safety concerns. In addition, these high-energy particles may also collide with the atmosphere to generate many free electrons, which could absorb radio waves, thereby affecting radio communications. In this thesis, the main focus is on the effects of SPEs on space-born electronic applications. The SPE could be two to five orders of magnitude higher than the background conditions. Thus, the SPEs could be strong enough to cause hazards in space applications, detailed in Section 2.2. For example, the measured soft error rate of a 4k\*32 bit 0.25  $\mu\text{m}$  CMOS SRAM module in a geostationary satellite during a few SPEs is illustrated in Table 2.1 [DLH<sup>+</sup>07]. The approximate value of the background upset rate is obtained by linear fitting to the monthly average data. Because the error rate of space-born applications might skyrocket during the eruption of high-energy particles, it's critical to monitor the space radiation environment and forecast SPEs in order to protect the system early.



**Figure 2.4:** Tree structure of IC reliability failure mechanism and corresponding fault classification. The green block represents the main focus of the dissertation.

## 2.2 Radiation Effects in Integrated Circuit

The Integrated Circuit (IC) is defined as a collection of electronic circuits on a single tiny chip of the semiconductor material. IC technology (such as CMOS, BiCMOS, Silicon-On-Insulator (SOI), bipolar, and others) has been able to make constant progress toward smaller device sizes, higher circuit densities, and faster-operating speeds because of the development of novel materials and device technologies. These modifications, however, have significant ramifications for system reliability. Radiation impacts are one of the most critical factors impacting IC reliability. In fact, one of the key issues in the design of current nanoscale ICs for hostile radiation environments, such as space, is radiation-induced effects [EPE11]. This section discusses the radiation-induced effect, which could lead ICs to failure, especially for ICs produced in CMOS technology.

Figure 2.4 presents the categorization of radiation effects of ICs. Generally, there are two types of radiation impacts in electronics [RDS<sup>+</sup>08]: (1) transient effects, such as bit flip in the memory cell; and (2) long-term effects, such as threshold voltage shifts and mobility degradation. The transient effect is mainly composed of a variety of soft errors in SEEs. The SEE is created by a single high-energy particle and can assume many various forms, and cause non-destructive errors (i.e., soft errors) or destructive errors (i.e., hard errors) in ICs. The introduction of SEEs is detailed in Section 2.2.1. Aside from hard errors in SEE, Total Ionizing Dose (TID) and displacement damage are two more prevalent types of long-term radiation impacts on ICs. Furthermore, several physical processes that cause long-term radiation impacts, such as bias temperature instability, are linked to long-term electrical deterioration.

Displacement damage is a cumulative effect, usually caused by nuclear interactions, and leading to lattice defects [MHB13]. Displacement damage may cause a decrease in the gain and leakage current of bipolar transistors, thereby reducing the minority carrier lifetime. When charged particles pass through matter, the particle energy could be dissipated by exciting orbital electrons and elastic collisions with the matter nucleus. Therefore, collisions between the incident particles

and lattice atoms then displace the atoms from their original lattice positions. In space, protons of all energies and electrons and neutrons with energies above 150 keV could produce displacement damage. Moreover, the displacement damage is the non-ionizing damage due to the long-term accumulation of ionizing radiation.

The TID, or absorbed dose  $D$ , is the long-term cumulative effect in ICs mainly caused by electrons, protons, and gamma rays [TEC08]. As an ionization-inducing effect, energetic particles can cause charge trapping or form interface states at semiconductor-insulator boundaries (e.g., SiO<sub>2</sub>), affecting component behavior or material properties. Furthermore, in Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) devices, trapped charges can cause gate threshold voltage shifts, while in general semiconductors, interface states can significantly increase device leakage current. As a result, this phenomenon can cause an increase in current flow and result in permanent damage of a device. TID primarily affects the insulators (oxide layers) of ICs and has no significant effect on conductors and semiconductors. Usually, higher temperatures are used for annealing, such as temperatures around 100°C [SKH11]. However, annealing provides only partial recovery from TID.

As mentioned earlier, TID and displacement damage are cumulative ionization effects that can cause permanent damage to the IC, which is not the focus of this thesis. In this study, the main focus is on the SEE, especially the transient effects in the SEE, which are crucial for designing adaptive resilient systems.

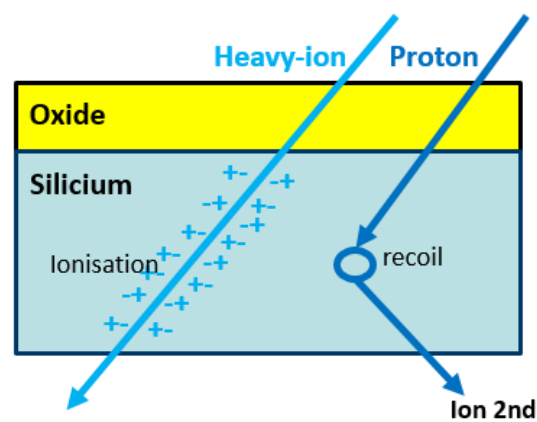
### 2.2.1 Single-Event Effect

The SEEs have become the major radiation-induced effect for space-borne applications as technology advances into the deep nanoscale range [EPE11]. A single high-energy ionizing particle penetrating sensitive nodes of an electronic device causes SEEs, which can result in apparently random failures in electronic systems. Due to SEEs, various error types, from temporary malfunction system responses to catastrophic system failures, could occur. As a result, effective SEE mitigation methods in electronic systems for space applications are necessary. Furthermore, this problem is highly crucial because of its unpredictability, the ubiquitous spectrum of high-energy space particles, and the rising sensitivity of electronics to SEEs as technology scales. Protons and heavy-ions, according to [NAD<sup>+</sup>14], are the principal sources of producing SEEs in semiconductor technology prior to 14 nm. Due to the lower critical charge of the advanced semiconductor technology generation, more types of particles (such as muons, neutrons, electrons, etc.) have also become essential sources of SEE.

Figure 2.5 illustrates the effect of high-energy particles striking a semiconductor to induce ionization. This thesis focuses on space-borne applications, mainly considering indirect ionization (i.e., proton induction) and direct ionization (mainly heavy-ion induction), other types of particles can be ignored [EPE11]. A single heavy-ion could pass through the silicon of a memory cell and deposit a track of ions, thus, resulting in an SEE. Heavy-ions can be energetic enough to pass



through memory cells in space. If these resulting ions are in the presence of the natural or applied field in an electronic device, they are gathered at the device electrodes. This generates an electric pulse or signal, which the device may perceive as a command to which it must respond. SEEs can also be initiated by protons, although the effect is normally not caused by the proton's direct passage. The proton produces just a little amount of ionization. In the silicon device, one proton in every  $10^5$  would undergo a nuclear event, and these events can produce heavy-ions with enough energy to cause disruption [EPE11]. Despite the appearance of a tiny number of occurrences, protons in the proton radiation belts (i.e., the Van Allen belt) are strong enough in space to cause many more disturbances than heavy-ion cosmic rays in the same environment.



**Figure 2.5:** Simplified ionization mechanisms responsible for radiation-induced faults: direct (left) and indirect (right) ionization.

Errors caused by SEE can be categorized into soft errors and hard errors. The soft error is a type of error that results from a signal or datum fault caused by transient SEEs. Soft errors do not damage the system's hardware, but can damage data or signals that are or will be processed. Soft errors can also be fixed by cold booting the system or rewriting the memory portion. Hard errors, on the other hand, are not recoverable since SEEs can cause destructive damage to transistors. According to [ASI14], the most important subgroups of SEEs in soft and hard errors are outlined as follows:

1. Hard SEEs (destructive):
  - Single-Event Latch-up (SEL)
  - Single-Event Burnout (SEB)
  - Single-Event Gate Rupture (SEGR)
  - Single-Event Gate Damage (SEGD)
  - Single-Event Dielectric Rupture (SEDR)
2. Soft SEEs (non-destructive):
  - Single-Event Upset (SEU)

- Single-Event Transient (SET)
- Single-Event Latch-up (SEL)
- Single-Event Functional Interrupt (SEFI)

Single-Event Latch-up (SEL) is an abnormal high-current state event, which is caused by the creation of a low-impedance path as a single high-energy particle traverses a sensitive region of the circuit. If the device is not powered off in time, large currents generated from SEL could overheat and damage the semiconductors. Thus, SELs could be destructive or non-destructive. If the semiconductor is not damaged, the power cycling (turning off and on) method can be applied to correct the effect.

Single-event burnout (SEB) and single-event gate rupture (SEGR) are two catastrophic consequences of power MOSFETs that might cause failure in space systems. Heavy-ions striking the device in the off state and forming electron-hole pairs along their routes trigger the failure process in both circumstances. The ions impact the device's neck area or the region between the p-base diffusions on the surface of an n-channel power MOSFET in the case of SEGR. Furthermore, before SEGR, when the gate-oxide is ruptured and begins to conduct, Single-Event Gate Damage (SEGD) might be noticed. A related phenomena, Single-Event Dielectric Rupture (SEDR), which is the development of a conducting route initiated by a single ionizing particle in a high-field area of a dielectric, may also be observed in CMOS devices.

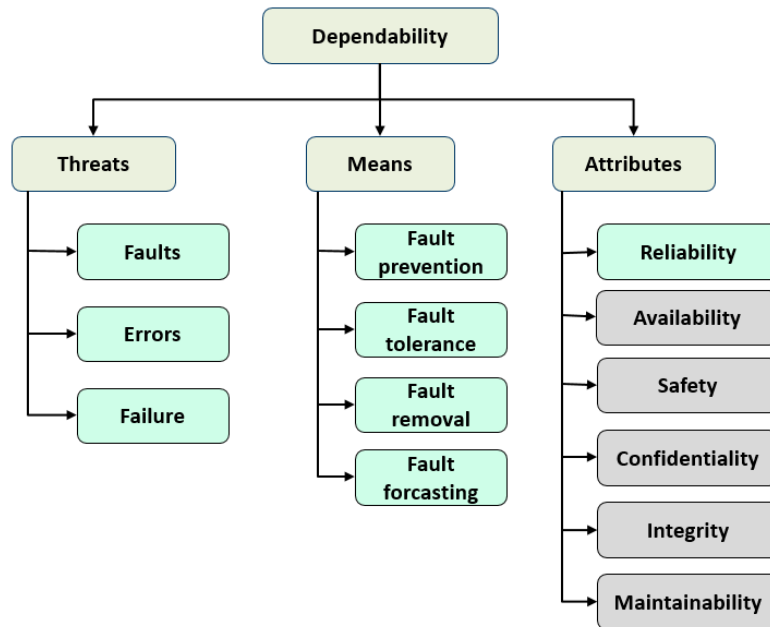
Single-event upsets (SEUs) are a key subset of single-event upsets (SEEs) in contemporary nanoscale CMOS integrated circuit architecture for space applications. SEU occurs when an energetic particle travels across the sensitive area of an off-state transistor, causing a transient failure in storage components. Charge deposition occurs when energetic particles pass through. The essential need for an SEU to occur is that the deposited charge exceeds the element's critical charge. SEUs can occur when a particle directly hits a memory element or when a particle-induced glitch in combinational logic propagates across and is caught by a memory element. While SEU refers to a single bit, it can be further broken down into the following subcategories:

- Single-Bit Upset (SBU): the SBU is equal to the SEU, i.e., single-bit flip, usually used to distinguish it from Multiple-Bit Upset (MBU).
- Multiple-Bit Upset (MBU): the MBU refers to the corruption of a set of adjacent bits in a digital element caused by a single particle.
- Multiple-Cell Upset (MCU): the MCU is a special case of MBU, which is a corruption of multiple physical adjacent bits, but these bit flips do not belong to the same logic unit, such as the physical adjacent cells in the interleaved memory [CSB<sup>+</sup>15].
- Single-word Multiple-bit Upset (SMU): the SMU is a special case of MBU in which energetic particles induce upset more than one bit belonging to a single word [KRL<sup>+</sup>93].

When an energetic particle impacts a sensitive transistor and deposits a certain charge over the associated critical charge, a single-event transient (SET) occurs at the output of a logic cell as a

voltage glitch. Moreover, in order to induce a soft error, the SET pulse needs to propagate through subsequent logic gates and be latched by memory or sequential elements.

A Single Event Function Interrupt (SEFI) is a soft fault that causes a component to stop working properly for some amount of time, such as a reset, deadlock, or another observable defect. In addition, SEFI usually occurs in critical parts of the component and can usually be recovered by reset, power cycling, rewriting control registers, etc.



**Figure 2.6:** Dependability-related terms are arranged in a tree. The green blocks represent the main focus of this dissertation.

### 2.3 Dependability of System

The design of resilient electronic systems is critical for devices operating in harsh environments, such as space. The resilient of a computer system refers to the ability of the system to be reasonably relied upon for the services it provides [EDU13], i.e. to maintain the dependability of the system against performance degradation or functional impairment due to various faults or anomalies. In other words, a system's dependability is determined by how much users believe in the accuracy of its outputs. According to [ALB<sup>+</sup>04], attributes, threats, and means are three core aspects that play an essential role in dependability and impact faith in the correctness and continuation of the product. They are depicted in Figure 2.6 and are discussed in depth in the following subsections.



**Figure 2.7:** Failure propagation sequence.

### 2.3.1 Threats to dependability

Faults, errors, and failures are three main threats to system dependability. In general, a fault occurs at the physical level caused by some failure mechanism, an error is an effect on a subsystem level, and a failure can produce systemic effects [EDU13]. The details of these threats are introduced as follows:

- **Fault:** a fault is the determined or suspected source of an error in hardware structures or software modules [ASI14]. When it comes to persistence, faults can be divided into three categories: permanent faults caused by aging, manufacturing defects, wear-out, etc.; transient faults caused by external environments, such as radiation-induced SEEs; and intermittent faults that occur under certain conditions and disappear when those conditions change, such as ground bounce, crosstalk, etc. Active faults could generate errors, while dormant faults do not affect the system.
- **Error:** an error is a visible failure in the system, and at least one subsystem has the failure [ASI14]. As a result, an error can be defined as inaccurate behavior that would be noticed within the system but may not be visible outside the system. The precondition for an error is a fault, that can (but must not) exhibit as an error.
- **Failure:** a failure occurs when errors propagate to the top-level of the system and affect the correct system outputs [ASI14]. Failure is a transition from correct to erroneous operation that can result in a variety of repercussions such as (temporary) loss of function, wrong control flow operation, etc.

As a result, as illustrated in Figure 2.7, a failure is caused by an error, and an error is a fault which is visible. As introduced in [ALB<sup>+</sup>04], faults are dormant, unless an error is generated, an error is propagated only by active computational processes, and a failure only occurs when errors propagate to the top-level of the system.

### 2.3.2 Means to achieve dependability

Means, illustrated in Figure 2.6, refer to the methods or techniques which can be used to increase the dependability of systems. As described in Section 2.3.1, faults are the source of errors and failures. Therefore, fault mitigation is the critical means to achieve system dependability, and the general methods are described as follows [EDU13]:

- **Fault prevention:** this approach aims to prevent the occurrence or introduction of faults in the system. This can be achieved by using a variety of protections at the design, fabrication, and operating stages, such as applying rad-hard cell libraries, shielding, rigorous maintenance procedures, etc.
- **Fault tolerance:** this approach is applied to allow the system to continue performing correct tasks after the occurrence of faults. The general approach is to mitigate failures at different system levels through the use of redundancy, which allows for failure detection or recovery.

Both detection and recovery can take place when the system is running normally or while it is suspended.

- **Fault removal:** this approach is intended to remove system faults during system development and operating phases. During the system development stage, the system verification with corresponding diagnosis and correction techniques can be applied. Corrective or preventative maintenance, on the other hand, is used to remove faults while the machine is in operation.
- **Fault forecasting:** this method involves a qualitative and quantitative assessment of present faults in order to forecast potential faults as well as the implications of such faults over the next temporal period. Fault injection is a frequently used approach for assessing and forecasting the effect of defects in a system while it is in use.

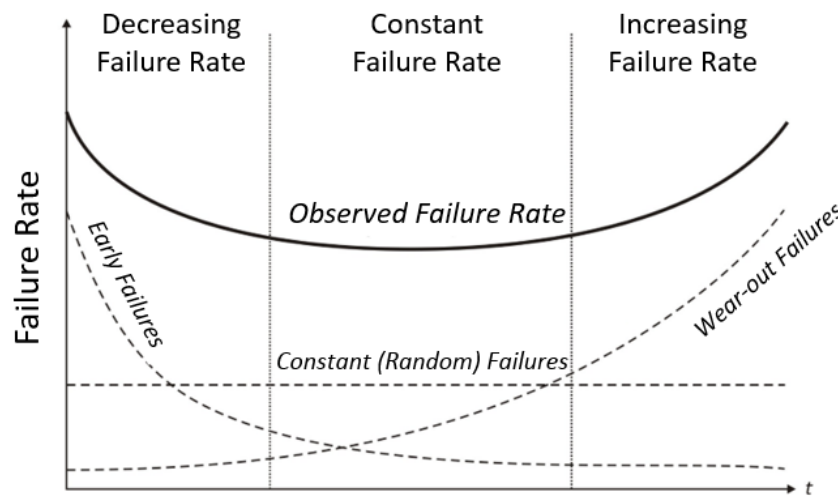
Although different means of designing a dependable system are described above, the most commonly used and also applied in this thesis to develop a self-adaptive resilient system is fault tolerance. In addition, this thesis also addresses methods for estimating and mitigating the soft error rate of the system module, and analyzes the influence on the dependability, especially the reliability, of the target system.

### **Fault Tolerance**

As mentioned above, fault tolerance is a method of exploiting and managing redundancy to mask failures that occur. There are a range of redundancy approaches that may be used at different system level [ECH<sup>+</sup>16], and different fault tolerance techniques are classed based on the kind of controlled redundancy:

- **Hardware redundancy:** the N-Modular Redundant (NMR) system, which provides two or more physical copies of hardware components, is the most common method of hardware redundancy. It can be further divided into three types, based on the status of the activated redundancy module: passive schemes, which use static space-redundancy methods to mask faults; dynamic schemes, which can switch to spare components when faults are detected; and hybrid schemes, which combine static and dynamic operation. Hardware redundancy is a reliable and effective system design option. However, the hardware redundancy method could also bring several penalties, such as system area, power, performance, time to design, etc.
- **Software redundancy:** with the software redundancy method, multiple programs can be used to compute the same function. Moreover, these redundant software are developed independently of each other. The target system runs several of these programs and compares the output. Thus, the correctness of the target software calculation results can be ensured. However, this method is mainly used to address software faults and may not be able to effectively avoid faults caused by hardware.

- Time redundancy: the approach of time redundancy is to do the same calculation several times using the same hardware components. This method is similar to software redundancy but runs the same software during redundant execution. As a result, time redundancy is mainly used to prevent temporary faults. Furthermore, this comes at the cost of at least double execution time.
- Information redundancy: information redundancy is based on data redundancy and is most commonly used to identify data transfer or storage mistakes. The Error Correction Code (ECC) approach is one of the most extensively used information redundancy technology, which adds additional parity information to the original data to achieve data correction or detection. Different types of ECC methods are widely used to protect memory and processor registers from transient faults, detailed in Section 4.2.2.



**Figure 2.8:** Bathtub curve representing failure rate [ASI14].

### 2.3.3 Attributes of dependability

The attributes for quantifying the negative influence (i.e., threats) and positive impact (i.e., means) on the system dependability are presented in the dependability tree in Figure 2.6. A brief description of these attribute properties is as follows [ASI14]:

- Reliability: continuity of correct operation
- Availability: the quality or state of being correct operation
- Safety: the condition of being safe from catastrophic consequences
- Confidentiality: absence of unauthorized disclosure of information
- Integrity: absence of inappropriate system state changes
- Maintainability: ability to maintain the current state of the system

The most critical property to quantify the impact of fault tolerance methods in this thesis is reliability, which is discussed in more detail in the following subsections.

### Reliability

The reliability,  $R$ , of electronic systems often relates to individual hardware units and reflects the probability that a unit will work successfully within a given time period. In general, the system reliability is used to describe the impact of fault-tolerance mechanisms on systems with a specific mission time. Reliability as a function of time  $R(t)$  represents the probability that the system will run successfully under specified conditions for a certain amount of time. Thus,  $R(t) = P(TTF > t)$ , where  $TTF$  represents a continuous random variable that specifies the time-to-failure. On the other hand,  $F(t) = P(TTF \leq t)$  gives the probability that the system would fail by time  $t$ . It is obvious that

$$P(TTF > t) + P(TTF \leq t) = R(t) + F(t) = 1.$$

Thus,

$$R(t) = 1 - F(t) = 1 - \int_{-\infty}^t f(s)ds = \int_t^{\infty} f(s)ds \quad (2.1)$$

where  $f(s)$  is the Probability Density Function (PDF) of system failure. Furthermore, if  $TTF$  is supposed to be exponentially distributed, empirical results demonstrate good agreement with measurements [AS114]. Thus, the PDF of system failure can be expressed as  $f(t) = \lambda e^{-\lambda s}$ . Therefore, the system reliability function is:

$$R(t) = \int_t^{\infty} \lambda e^{-\lambda s} ds = e^{-\lambda t} \quad (2.2)$$

where the  $\lambda$  stands for the system failure rate. Besides that, real systems frequently exhibit failure rate behavior similar to the bathtub curve observed in Figure 2.8. The bathtub curve is divided into three phases: the infant mortality phase, where the failure rate decreases; the useful operating phase, with a constant and long time period of low failure rate (usually several years for electronic equipment); the wear-out phase, where the failure rate increases mainly due to the aging effects. Moreover, due to the independent probability rule, the overall reliability of a system without redundancy can be expressed as the product of the sub-factors.

## 2.4 Supervised Machine Learning

The section introduces methods of supervised machine learning method, which is applied for the space environment prediction in Chapter 5. Machine Learning (ML) is the field of teaching computers to learn from data and fit it into models that people can understand and utilize. Since ML focuses on analyzing and interpreting patterns and structures in data using various algorithms, it is important in enabling learning, prediction, reasoning, and decision-making in various domains [TMM97]. There are many different types of machine learning systems, and they may be generally

divided into four groups based on whether they were trained with human supervision: supervised, unsupervised, semi-supervised, and reinforcement learning. This thesis mainly applies supervised learning methods, which is the most common branch of machine learning, and aims to learn by example. The supervised learning model is trained using "well-labeled" training data, and based on this data, the model can predict the output. Supervised learning can be split into the following two subcategories:

- **Classification:** classification is the process of determining a function that divides a dataset into groups depending on several characteristics and is used to categorize discrete values.
- **Regression:** the process of determining correlations between dependent and independent variables in order to predict continuous values is known as regression.

This thesis applies regression methods to continuous in-flight space radiation predictions based on historical space flux datasets as well as detected data, detailed in Chapter 5. In addition to the regression method, the online learning, dedicated hardware accelerator, and model evaluation methods are also addressed.

## Regression

As mentioned above, regression is a supervised learning technique for numerical prediction and is mainly used in predictive analytics to forecast trends and predict outcomes. Regression is the critical method of predictive modelling. Thus, the regression method is widely used to predict output, forecast data, analyze time series, and discover causal relationships between variables [FPE<sup>+</sup>11], such as predicting housing prices, stock prices, test scores, etc. The equation for basic linear regression can be written as below:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \dots + w[i] * x[i] + b \quad (2.3)$$

where  $\hat{y}$  is the predicted value,  $i$  is the number of features,  $x[i]$  is the  $i^{th}$  features value,  $w[i]$  is the  $i^{th}$  features weights, and  $b$  is the bias term. There are several types of regression algorithms, such as linear regression, logistic regression, multiple linear regression, decision tree regression, neural networks, etc. In this thesis, five existing regression models are used to achieve time series regression, which is a statistical regression approach for predicting a future reaction based on past responses and is discussed in Section 5.3.2.

It is worth noting that traditional regression learning techniques operate in batch mode, i.e., require complete training data as input. Therefore, the training method requires complete training data before learning the task, and this process is usually performed offline due to the high cost of training. Traditional regression methods have inefficiencies in time and space costs, poor scalability for large applications, inability to adapt to online dynamic variations, etc.



### **Online machine learning**

Online learning is a combination of different techniques in ML to train the system incrementally by providing data instances in sequence, i.e., the mini-batches approach. Each learning step is fast and cheap, and each step updates the best predictor for future data. Therefore, the system can learn new data in real-time and overcome the shortcomings of pure offline learning. Furthermore, online learning is well-suited for systems that receive data in a continuous stream, such as real-time space flux data, and can adapt to changes quickly and autonomously. Furthermore, the learning rate used is critical for online machine learning since it dictates how quickly the model responds to changing input. A high learning rate, in general, allows the model to quickly adapt to new input, while simultaneously quickly forgetting prior data. A low learning rate, on the other hand, can make the system more inertial, while also making it less susceptible to noise in fresh data or sequences of non-representative data points [GAU19].

Online machine learning methods also face some challenges. One of the key challenges is feeding bad real-time data into the model, which could gradually degrade the prediction performance. To reduce this risk, input data-providing systems, such as on-chip monitors, should be closely monitored and react to abnormal data. In addition, it is also important to provide a good initialization for the online machine learning model, otherwise, the convergence time could be long, and it may also produce unwanted output fluctuations.

### **Hardware accelerator**

Hardware accelerators are dedicated computer hardware designed to perform specific functions, which can provide more benefits (such as efficiency) than running the same functions on a general-purpose processor [STW11]. Unlike traditional processing systems that process instructions one by one and need to move temporary results in and out of storage elements, hardware accelerators can improve the execution of specific algorithms by allowing higher concurrency and reducing instruction overhead. Hardware accelerators can be generally implemented in two types of approaches:

- Custom logic: a dedicated electronic circuit can be designed to implement the accelerator's functionality as a hardware accelerator. The advantage of this method is that it can meet the balance of performance and other goals, and there is no limit to the realization of the acceleration function
- Programmable logic: programmable logic, such as Field-Programmable Gate Array (FPGA), is also commonly used as hardware accelerator. The main advantage of using programmable logic is that it is widely available and quickly applied to different accelerators, thus enabling rapid customization with high quality.

## Model evaluation

Model evaluation is the process of analyzing the quality of predictions made by a machine learning model. This process is critical to examining the efficacy of the trained model, and it also plays an important role in model monitoring. For evaluation, the trained machine learning model should be validated against a new independent dataset [GAU19]. By comparing the labelled reference results with the model's predictions, the following questions of the trained model can be answered: how about the model performance; is the model accurate enough; is the model under-/over-fitting? For regression machine learning models, there are several evaluation metrics:

- Mean Absolute Error (MAE): MAE represents the mean of the absolute difference between the actual and predicted values in the dataset, and thus, it measures the mean of the residuals in the dataset:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}| \quad (2.4)$$

where  $N$  represents the number of instances in the dataset,  $y_i$  is the true labelled reference result for instance  $i$ ,  $\hat{y}$  is the corresponding predicted value from the model.

- Mean Squared Error: (MSE): MSE represents the average of the squared differences between the true and predicted values in the dataset, thus, it can measure the variance of the residuals.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2 \quad (2.5)$$

- Root Mean Square Error (RMSE): RMSE is the square root of the mean squared error, and it can measure the standard deviation of the residuals.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2} \quad (2.6)$$

- Coefficient of determination or R-squared ( $R^2$ ):  $R^2$  represents the proportion of the variation in the dependent variable that is predictable from the independent variable, and it is a scale-free fraction, i.e. the value of  $R^2$  would be less than 1 regardless of whether the value is small or large.

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y})^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (2.7)$$

where  $\bar{y}$  is the overall mean.

- Adjusted R squared ( $R^2_{adj}$ ):  $R^2_{adj}$  is a modified version of  $R^2$ , which is adjusted for the number of independent variables in the model and is always less than or equal to  $R^2$ .

$$R^2_{adj} = 1 - \frac{(1-R^2)(N-1)}{(N-K-1)} \quad (2.8)$$

where  $K$  is the number of the independent variables in the target dataset.

## 2.5 Summary

This chapter provides the fundamentals for understanding further approaches of this thesis w.r.t. developing self-adaptive resilient processing systems for space applications. The space radiation environment is complicated and dynamic, and various events, including SPE, can produce a huge number of high-energy particles. Radiation particles, particularly SEEs, which are the major source of radiation-induced faults in space electronics, can cause destructive or non-destructive failures in electronics. Furthermore, the intensity of cosmic radiation during SPEs can change by five orders of magnitude in a matter of hours or days, increasing the chance of SEEs in space-borne electronic applications by several orders of magnitude. Moreover, system's dependability and corresponding characteristics are introduced in detail. Dependability describes the ability of a system to provide its expected level of service under threats (i.e., faults, errors, and failures), and the most critical dependability attribute considered in this thesis is reliability. Besides that, machine learning, especially the regression method, is introduced, and it is important to achieve the learning and forecasting of the space radiation environment. Moreover, the online learning, hardware accelerator, and model evaluation methods are addressed, which are also essential for designing a high-accuracy prediction model in processing systems.

## Chapter 3

### State of the Art

The increasing rate of IC faults necessitates the development of novel solutions that enable high system reliability. This chapter makes an in-depth overview of the state-of-the-art concerning the three main topics addressed in the self-adaptive resilience system design processes: (1) particle detection for the real-time radiation-induced faults monitoring, which is critical for the application in the complex radiation environment; (2) radiation environment and the system reliability behavior evaluation and prediction method, which is necessary to provide the early protection of the target reliability-critical system; (3) dynamic radiation-induced fault mitigation, especially for self-adaptive fault tolerance systems, which is essential for performing the run-time optimal online mode dynamic selection method on the target system.

This chapter is divided into four sections. Section 3.1 reviews the existing solutions for the existing particle detectors. Section 3.2 introduces the prediction or analysis methods of the space environment and system reliability. Section 3.3 is devoted to techniques for dynamic mitigation methods regarding reliability, lifetime, performance, power consumption, etc. All sections outline the benefits and drawbacks of each option under consideration. Moreover, section 3.4 summarizes the advance that proposed works make beyond the state-of-the-art.

#### 3.1 Particle Detector

Radiation-induced faults are among the most common causes of failure in space ICs. Design phase hardening approaches can generally effectively mitigate radiation-induced faults in a complicated IC design. However, if the static mitigations at the design phase are based on worst-case radiation scenarios, the resulting system overhead (such as area, power consumption, performance, etc.) will be extremely high and unnecessary most of the time. A cost-effective alternative is to use adaptive fault-tolerance techniques, which involve activating the appropriate fault-tolerant mechanisms based on the current radiation environment. In that regard, the sensory hardware for the space particles, which can monitor the system error rates, is necessary. As introduced in Section 2.2.1, SEEs can be induced by high-energy particles. Thus, in order to give a full evaluation of the

radiation environment, a particle detector should monitor the flux and intensity of radiation particles. In addition, the detector should be more sensitive to particle impacts than the standard logic gates that make up the target design, which can guarantee early warning of the target system. Moreover, the particle detector should have a large enough sensing area to ensure high particle sensitivity. Furthermore, in-situ particle monitoring, which may be incorporated on the same chip as the target system, is critical for adaptive fault-tolerance applications. Thus, the particle detector can sense the same radiation conditions as the exposing of the target system. Moreover, in order to maintain the overall system area and power consumption, particle detectors should also be carefully designed and planned as well as placed in the target system.

For space applications, there are numerous types of semiconductor-based particle detectors. Based on the design and operating principles, particle detectors can be divided into six basic categories [ACS<sup>+</sup>20]: (1) bulk built-in current detectors, (2) acoustic wave detectors, (3) diode detectors, (4) 3D NAND flash detectors, (5) pulse stretching inverter detector and (6) Random Access Memory (RAM)-based detectors. Each form of particle detector has benefits and disadvantages, which are addressed in further detail in the subsections that follow.

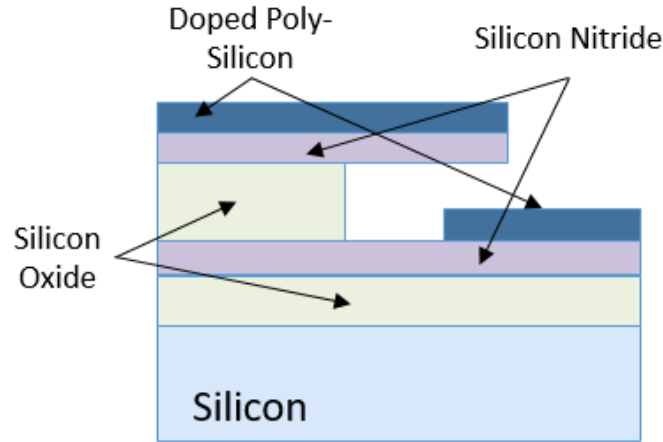
### 3.1.1 Bulk Build-in Current Detectors

As introduced in Section 2.2, transient current pulses may be caused by the energetic particle. Therefore, the current sensors can be applied for detecting these events in the target semiconductor device. Moreover, in order to perform particle strike detection in combinational logic, the Bulk Built-in Current Sensor (BBICS) can be applied [NRV<sup>+</sup>05] [GWF07]. When the bulk current reaches a certain threshold, the detector creates a flag signal that may be used to trigger the soft error correction mechanism. BBICSs are similar to PMOS and NMOS transistors in that they are directly linked to the transistor bulk terminals. Figure 3.1 shows an example of a basic PMOS-BBICS system made up of three transistors. The NMOS-BBICS has a similar construction, with PMOS transistors replaced with NMOS transistors. When the high current level surpasses the threshold, the detector emits a flag signal, which activates a soft error-correcting process.

Since BBICS are directly connected to the target circuit, one of the main advantages is their ability to provide accurate fault location information. Thus, the system with BBICS can enable the error correction mechanism to be activated only within the affected sub-circuits. Another essential property of BBICSs is their low power consumption, which is mostly due to leakage power. Besides that, since one detector can be utilized to monitor multiple transistors [BGT<sup>+</sup>18] [RPB<sup>+</sup>20], the number and spatial distribution of BBICSs in the system be appropriately designed to achieve the trade-off between the detection efficiency and costs.

However, BBICS has several limits in its use. The primary issue with the BBICS implementation is that, while the impact location in the target circuit can be determined, particle flow data cannot be directly collected. Thus, it could be hard to directly analyze the corresponding radiation flux threat on the entire system. In order to perform the impact of dependability on the





**Figure 3.2:** Typical cantilever sensing structure for acoustic wave detector [UVG16].

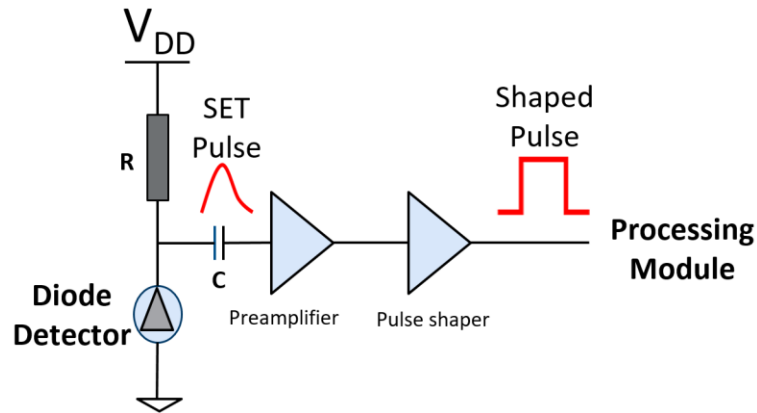
As previously stated, many acoustic wave detectors must be dispersed over the target chip in order to achieve high-accuracy detection for radiation-induced faults. By analyzing the relative time difference between the arrivals of acoustic waves from different detectors, the particle strike location can be accurately obtained [UVG16]. However, similar to the BBICS designs, the acoustic wave detector could also be prone to false alarms from non-radiation sources. Furthermore, for sensing the system in the self-adaptive processing system, more complicated processing circuitry with a large number of detectors across the chip is required.

### 3.1.3 Diode Detectors

One of the most common particle detectors is the diode-based p-n junction detector. They are available in a variety of configurations, including strip detector [SKA<sup>+</sup>09], active pixel detector [WSW<sup>+</sup>20], scintillator-coupled detector [SRO<sup>+</sup>19], etc. These detectors are frequently constructed with a reverse bias to obtain minimum leakage current and maximum depletion layer width, resulting in high detection efficiency. Depending on the particle intensity, radiation particles can create continuous or pulsed current pulses in the detector. As a result, measuring radiation-induced current may offer detailed information on radiation exposure as well as correctly measure induced charge, particle LET, flux, and energy spectrum.

However, if diode-based detectors were used in an adaptive processing system to trigger a dynamic fault tolerance mechanism, the cost would be prohibitively high due to the necessity to integrate several approaches. Diodes are not often manufactured with the same technology as the target device, making detector integration challenging. For diode detectors, a mixed-signal processing circuit is also required, raising the total cost and design complexity of the target system. Furthermore, in order to analyze the detected signals, the diode detector requires specialized processing circuitry, which is typically comprised of a preamplifier, a pulse shaper, and an analog-to-digital converter, as shown in Figure 3.3. Due to the real-world implementation may include

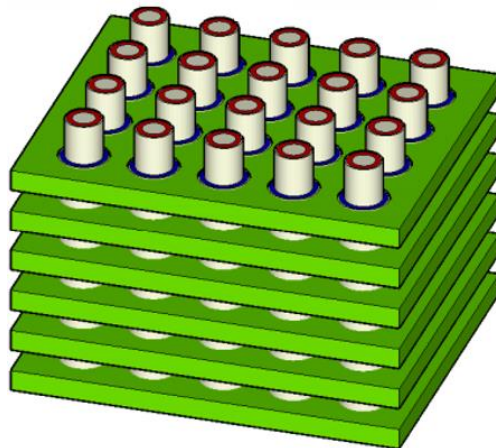
hundreds or thousands of diodes on the same substrate, the hardware and power overhead generated by the processing logic could be enormous.



**Figure 3.3:** Example of a processing channel for a diode-based detector [ACS<sup>+</sup>20].

### 3.1.4 3D NAND Flash Detector

The application of 3D NAND flash memory with floating gate transistors as a heavy-ion detector is presented in [MBA<sup>+</sup>20]. Figure 3.4 shows an architecture of a 3D NAND array from [MBA<sup>+</sup>20], where the cells are organized in vertical columns with a vertical channel structure, and the word lines are horizontal planes orthogonal to the channels. The basic idea behind this concept is to detect energetic particles by detecting the threshold voltage shift generated by incoming particle charges deposited on a floating gate transistor. Because of the 3D structure of NAND flash memory, the error rate, the incident angle of incident particles, and the associated LET can all be monitored. Moreover, this 3D structure can also allow for distinguishing between errors caused by energetic particles and errors caused by electrical noise as well as distinguishing between SEUs and MBUs.



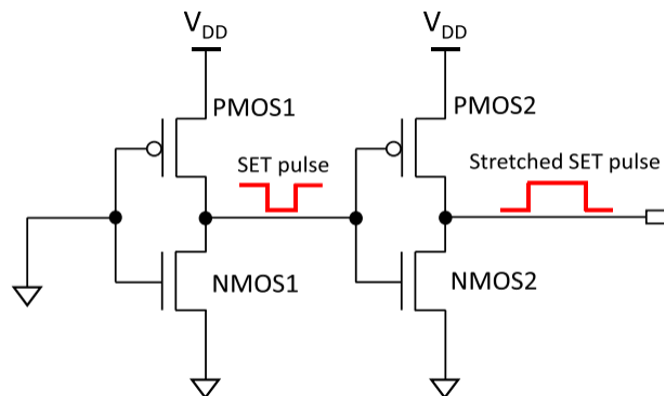
**Figure 3.4:** Example architecture of a 3D NAND floating gate array [MBA<sup>+</sup>20].



Although 3D NAND flash memory with floating gate transistors is a promising technology with considerable advantages in radiation particle detection, its present primary disadvantage is the difficulty of incorporating it into devices. It might be not very easy to integrate 3D NAND flash and floating gate technologies into a standard CMOS IC design. Furthermore, analogue processing circuits and analog-to-digital converters are required to monitor the variance of the threshold voltage of floating gate transistors precisely. As a result, the processing electronics for 3D NAND memory may be complicated and expensive.

### 3.1.5 Pulse Stretching Inverter Detector

The use of custom-sized pulse-stretching inverter chains as particle detectors was presented in [ACS<sup>+</sup>21]. The fundamental concept is to monitor the SET count rate and pulse width fluctuations. Thus, the appropriate particle flux can be calculated using the SET count rate, and the LET fluctuations can be calculated using the SET pulse width variations. Figure 3.5 presents the fundamental sensing element, i.e., Pulse-Stretching Cell (PSC), of this detector, consisting of two inverters connected in series. When the fixed logic level at the PSC input is set, two transistors are constantly active, while the other two are always off. Off-state transistors are vulnerable to particle effects, whereas on-state transistors function as restorative components. Furthermore, the PSCs are skewed in size, which implies that in one inverter, the PMOS transistor has a larger channel than the NMOS transistor, while the NMOS transistor has a wider channel than the PMOS transistor in the other. Off-state transistors should have as wide a channel width as possible in order to provide a sufficiently large sensing area of a PSC and therefore increase the likelihood of particle strikes. A parallel or serial design structure with PSG cells is also necessary to provide an appropriate sensing area [ACS<sup>+</sup>21]. The serial design is only useful for monitoring the SET count rate because the SET pulse width changes so little across such a large range of LET. In contrast, the parallel design allows for the collection of both SET count rate and SET pulse width fluctuation.

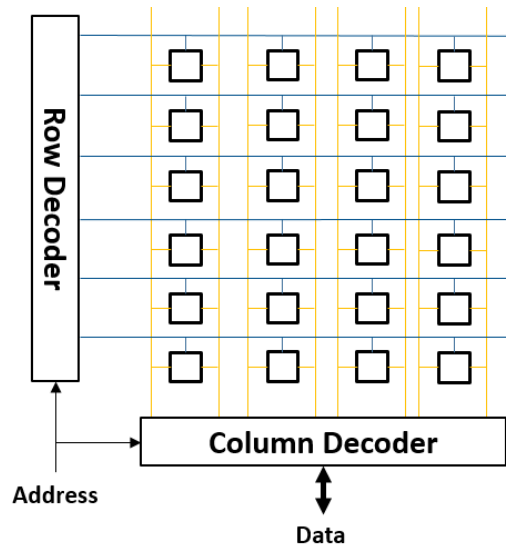


**Figure 3.5:** Basic structure of two-inverter Pulse-Stretching Cell (PSC) [ACS<sup>+</sup>21].

The pulse-stretching inverter detector appears to be a potential method for detecting particle-induced SETs and LETs. It should be emphasized that this method does not precisely measure the SET pulse width, but rather sorts the observed SET widths into different ranges. This is due to the fact that digital processing was chosen in this case as a simple and low-cost alternative to analog processing in diode detectors. Furthermore, in order to offer an appropriate sensing area, a large enough PSC array is required, which may result in an increase in chip cost. In addition, the processing logic is required to manage the outputs of all pulse-stretching arrays, such as a typical OR-tree [ACS<sup>+</sup>21], which incurs extra costs.

### 3.1.6 RAM-based Detectors

RAM (such as SRAM, BRAM, DRAM, etc.) is a computer memory that can be read and written in any order and is widely used to store binary data. The use of commercial or custom-designed RAMs, particularly SRAM, as particle detectors has shown to be an advantageous method for monitoring particle-induced faults, particularly SEUs, in a variety of terrestrial and space applications [ACS<sup>+</sup>20]. In Figure 3.6, the basic structure of the RAM array is shown, and the storage cells (i.e., the sensing part) are various for different types of RAMs. Due to advantages of low cost, high density, high sensitivity to radiations, and the possibility of implementation in different technologies, RAMs, especially SRAM, are widely used as particle detectors [HGR<sup>+</sup>05] [GMR<sup>+</sup>17] [TKK<sup>+</sup>17] [PTB<sup>+</sup>19]. The operation approach relies on counting the number of faults in RAM cells. The proper logic state is altered when a particle collides with a sensitive transistor within the cell and deposits energy higher than the critical charge. In general, detector sensitivity is related to RAM size, or the number of storage cells.



**Figure 3.6:** Simplified architecture of the RAM array. Hollow squares represent storage cells, which vary for different RAM types.

RAM-based detectors feature a simple working principle, do not require analog processing, are straightforward to integrate, have large sensing areas, and can be manufactured using the same technology as standard ICs. In general, there are two most common approaches to perform fault detection: 1) set and check pre-defined test patterns [HGR<sup>+</sup>05] [TKK<sup>+</sup>17] [PTB<sup>+</sup>19], and 2) ECC + scrubbing method [GMR<sup>+</sup>17]. Moreover, it is also possible to adjust the sensitivity of the storage cells by modifying the transistor structure [TKK<sup>+</sup>17], scaling the supply voltage [PTB<sup>+</sup>19], etc. Furthermore, the appropriate particle flow can be estimated based on the determined number of defects and experimentally obtained cross-section data of the target RAM. Besides that, compared to alternative systems, the appropriate digital circuitry for processing identified defects is simple to construct.

However, RAM-based detectors have several limitations. The main disadvantage is that the vast majority of designs are based on standalone implementation, in which the RAM blocks can only be used for the detection of radiation particles. Therefore, the size of the sensing area could be limited, and the area overhead due to the standalone implementation could also be considerable. Furthermore, owing to error detection methods, faults could not be communicated in real-time, and the reaction time of RAM-based detectors may be slower than that of other detectors. In addition, RAM-based detectors perform the particle-induced fault counts, thus, the corresponding LET variation cannot be detected. Moreover, since the RAM blocks could also be affected by multi-bit upsets or permanent faults, thus, the detection accuracy of traditional “ECC + scrubbing” methods could be impacted and have the over-counting issue.

### 3.1.7 Comparison of Particle Detectors

In Table 3.1, a comparative comparison of the strengths and weaknesses of introduced particle detectors is detailed. As introduced in the previous subsections, each particle detector design has different advantages and disadvantages, and designers need to make reasonable choices according to different application scenarios and requirements. Diode-based and RAM-based detectors are the most often utilized particle detectors for space applications [ACS<sup>+</sup>20]. However, none of the currently available designs can deliver ideal performance for self-adaptive resilient systems, as no state-of-the-art design can simultaneously fully satisfy the following requirements: 1) Sensitive to radiated particles, 2) Sufficiently large sensing area, 3) Flexible expansion, 4) Easy to integrate into target chip, 5) Low area and power consumption, 6) Accurate error detection, i.e., avoid over-counting from MBU and permanent failure, 7) Distinguish error types, 8) Easy to build corresponding processing circuit. The closest existing design to meeting all of the above requirements is from one BRAM-based SEU detector in reconfigurable FPGA [GMR<sup>+</sup>17], where a custom wrapper is designed with scrubbing and error detection methods for both free and used FPGA BRAM blocks. Therefore, the additional cost for the non-standalone BRAM-based detector blocks is low and easy to achieve in the FPGA designs. However, this design does not solve the potential over-counting issues, especially from permanent faults, thus, the detection accuracy

could be affected. In addition, the custom FPGA BRAM wrapper can be optimized, especially for multiprocessing systems and external RAM blocks. From the viewpoint of self-adaptive resilient applications, an ideal embedded particle monitor should trade-off precise detection of radiation exposure and low-complexity processing logic with low-area and low-power overhead, as well as the ability to identify various types of failures. Furthermore, in order to reduce total costs, the monitor should be able to be incorporated on the same chip as the target system. Due to unavailability of such a solution, further research has been conducted into revolutionary low-cost embedded particle detectors.

**Table 3.1:** Comparison of radiation particle detectors.

Type of Detector	Advantages	Disadvantages
Bulk built-in current detector	<ul style="list-style-type: none"> <li>• Detect the particle strike location</li> <li>• Easy to be integrated</li> </ul>	<ul style="list-style-type: none"> <li>• Sensitivity could deteriorate</li> <li>• Maybe prone to false alarms</li> <li>• Multiple detectors are required</li> </ul>
Acoustic wave detector	<ul style="list-style-type: none"> <li>• Detect the particle strike location</li> <li>• Easy to be integrated</li> </ul>	<ul style="list-style-type: none"> <li>• Complex analog processing circuit</li> <li>• Maybe prone to false alarms</li> <li>• Multiple detectors are required</li> </ul>
Diode detector	<ul style="list-style-type: none"> <li>• Measure particle flux, LET and energy spectrum</li> <li>• Detect the direction of particle incidence</li> </ul>	<ul style="list-style-type: none"> <li>• Complex analog processing circuit</li> <li>• Challenging for integration</li> <li>• Multiple detectors are required</li> </ul>
3D NAND flash detector	<ul style="list-style-type: none"> <li>• Measure particle flux and LET</li> <li>• Angle of particle incidence</li> <li>• Distinguish error type</li> </ul>	<ul style="list-style-type: none"> <li>• Complex analog processing circuit</li> <li>• Challenging for integration</li> </ul>
Pulse stretching inverter detector	<ul style="list-style-type: none"> <li>• Measure particle flux and LET</li> <li>• Easy to be integrated</li> </ul>	<ul style="list-style-type: none"> <li>• Multiple detectors are required</li> <li>• Additional processing circuit is necessary</li> </ul>
RAM-based detector	<ul style="list-style-type: none"> <li>• Large sensing area</li> <li>• Easy to be integrated</li> <li>• Distinguish error type</li> </ul>	<ul style="list-style-type: none"> <li>• Cannot measure LET</li> <li>• May have over-counting issue from MBU and permanent faults</li> </ul>

### 3.2 Space Weather and System Reliability Analysis

As introduced in Section 2.1, the space weather is complex and dynamic, and the corresponding radiation particle flux may fluctuate significantly. The Sun is one of the most prominent causes of this dynamic variety of the space environment. The Sun can initiate SPEs, which are high-energy, long-duration particle events, on an occasional basis, resulting in a highly dynamic radiation space

environment. As a result of changes in the space environment, the probability of faults for the target system can vary several orders of magnitude, thus, seriously affecting the system's reliability. Therefore, it is important for space applications to perform the early prediction of the space environment and the system reliability changes during run-time. This section introduces existing corresponding methods in following subsections.

### 3.2.1 Space Weather Recognition

There is a vast and freely available collection of space weather observations gathered over several decades by numerous space missions, which has several advantages for space weather study and prediction. One of the most well-known datasets comes from the Geostationary Operational Environmental Satellite (GOES) system, and it comprises observations of the geomagnetic field, particle fluxes, and X-ray irradiance at geostationary orbit since 1974 [GOES]. Several research satellites that monitor space weather operate in a halo orbit around the Sun-Earth L1 point (the Earth's neutral gravity point): The Advanced Composition Explorer (ACE) satellite [ACE] investigates matter composed of energetic particles from the solar wind and the interplanetary medium; the Solar and Heliospheric Observatory (SOHO) [SOHO] provides results of investigations into the Sun's outer layer, solar wind, and associated phenomena; and the Global Geospace Science (GGS) Wind satellite [WIND] has been investigating how the Sun's magnetic field is generated and structured, as well as how this stored magnetic energy; the Deep Space Climate Observatory (DSCOVR) satellite [DSO] was the National Oceanic and Atmospheric Administration's (NOAA) first operational deep space satellite, serving as the agency's principal mechanism for alerting Earth of solar magnetic storms. Furthermore, GPS satellite data has been made available to the public in recent years, providing a plethora of information on particle fluxes [MSC<sup>+</sup>17]. Aside from space-based observations, ground-based magnetometer arrays may detect changes in the Earth's magnetic field [EEZ17].

The NOAA [NOAA] plays one of the most important roles in measuring geomagnetic storm events, radio blackouts, and solar radiation storm events, i.e., SPEs. Moreover, NOAA also provides information on the impact on biological systems, technical systems, satellite operations, etc. The Coronal Mass Ejection (CME) scoreboard [CEN17] provides a central location for the community to submit, observe, and evaluate the prediction of CME phenomena, which produce SPEs. The best CME prediction model can foresee this phenomena hours in advance, using continuously accessible data, such as X-ray flux, radio burst data, related flare position, etc.

In the recent past, different machine learning algorithms have been employed to forecast space weather in space missions for various purposes, such as planning spacecraft and satellite routes/maneuvers, protecting astronauts, etc. The author of [ECA<sup>+</sup>19] reviewed the current achievements, forecasting opportunities, and future role of machine learning in space weather, particularly the prediction of geomagnetic indices and relativistic electrons at geosynchronous orbits, as well as the occurrence of solar flares, coronal mass ejection propagation time, and solar

wind speed. In [HMB<sup>+</sup>18], the authors applied the machine learning classification techniques to the existing Space Weather Prediction Center (SWPC) statistical proton prediction model, and the result showed that the machine learning model could make a much faster decision than the previous numerical models. A NASA-invest space intelligence system is proposed in [AJE<sup>+</sup>17], which has the capability to forecast solar-driven events and provides high fidelity and pre-to-post eruptive transitional forecasts.

However, space weather, and especially solar events, is still a challenging research topic and could not be reliably predicted, due to a few reasons [ECA<sup>+</sup>19], such as lack of information, gray-box problem, surrogate problem, uncertainty problem, irregular appearance, etc. According to several solar events prediction methods in [CEN17], there are no tools that can accurately forecast CMEs and SPEs a few days in advance but usually a few hours in advance. Besides that, it is also hard to predict if the solar event will strike the Earth and affect our environment. Moreover, the current quantitative accuracy prediction methods require a lot of continuous data, such as particle flux data, flare location, and radio burst data. Thus, it is challenging to perform real-time in-flight predictions on satellites.

Regarding space weather recognition in a self-adaptive resilient system, it is essential to provide real-time space environment detection and prediction, thus, suitable radiation protection techniques can be deployed before the burst of particles. In [MCA<sup>+</sup>09], onboard memories and FPGA primitives were used in experimental fly satellites to calculate SEU rates in space and concluded that measurements could be one to two orders of magnitude away from pessimistic calculations. But this design only realizes the static space environment detection without further analysis. The above method is extended in [GRH16] by analyzing the Mean Time To Upsets (MTTUs) of the detected faults, thus, solar conditions can be identified from changes in the MTU time interval. However, such a paradigm only allows for the classification and recognition of solar events into five static levels, making it impossible to monitor fine-grained real-time fluctuations of the SPE. Moreover, to the best of my knowledge, the on-chip fine-grained space environment detection and prediction has not been addressed by any existing designs.

### 3.2.2 System Reliability Analysis

As indicated in Section 2.3, aggressive technological growth trends have considerably amplified the occurrence and effect of failures in computer systems, particularly radiation-induced defects in electronic systems. Many academics have concentrated on the topic of quantitatively measuring system dependability, and it is critical for designers to avoid both under- and over-protection. One of the most important pieces of research is the Architectural Vulnerability Factor (AVF) [MWE<sup>+</sup>03], in which a method for evaluating the susceptibility to soft errors on full-system microprocessors has been introduced. The AVF is determined by keeping track of the bits in the structure required for successful execution, which means that any inaccuracy in these bits would result in a system output error. Besides the full-system reliability recognition, the work on

quantifying system reliability in each individual system stack layer has been performed. In [SRK09], a microarchitecture-independent method, Program Vulnerability Factor (PVF), is proposed to quantify the fault-masking inherent to a program. Furthermore, [SRK10] extends a similar approach to measure hardware vulnerability irrespective of program-level masking. In addition, in [SRK10], the authors develop the concept of "system vulnerability stack," which may compute the vulnerability factor of each layer of the system stack. A Register Vulnerability Factor (RVF) is proposed in [YWZ05] to evaluate methodologies that analyze the soft error probability in registers, impacting system dependability. This method employs profiling to guide register allocation, allowing ECC registers to have the highest RVF values, and boosting the register file's dependability. The authors of [BHJ10] developed an Instruction Vulnerability Factor (IVF), which measures the error possibility for instructions that affect program output. The IVF value can be determined via offline profiling with simulation or fault-injection experiments, allowing each instruction to be toughened at different reliability criteria during run-time.

In order to accurately and efficiently estimate corresponding system reliability metrics, three main estimation methods are commonly used: fault-injection campaigns [FUK12], offline-based analysis [WHG07], and run-time estimation [BRC<sup>+</sup>05]. The authors of [FUK12] ran fault injection experiments on three GPU structures to calculate the necessary AVFs: register file, local memory, and active mask stack. In [WHG07], the regression approach is used in an offline way to examine the link between AVF and other microarchitecture characteristics, which can aid in lightweight AVF prediction at run-time. In [BRC<sup>+</sup>05], a lifetime analysis method for assessing the AVF of address-based structures (such as data caches, data translation buffers, and so on) has been introduced, in which the bit level AVF may be estimated as a proportion of its lifetime. Furthermore, in [CRL<sup>+</sup>06], a simulation approach is described that can perform multiprocessing system reliability analysis, overcome the constraints of statistical and architectural-level models, and fill gaps by studying reliability at the core level.

Machine learning methods have also been used in the rad-hard system design flow, such as optimizing the Soft Error Rate (SER) characterization. F. Rocha de Rosa et al. [FRR<sup>+</sup>19] applied the supervised and unsupervised machine learning techniques to the multicore system for the purpose of soft error analysis. In [TLA<sup>+</sup>19], T. Lange et al. used machine learning algorithms to optimize the fault-injection simulation campaigns as well as evaluate the system functional failure rate. S. Hirokawa et al. [SHI<sup>+</sup>16] exploited the machine learning method to facilitate the soft error discrimination method and verified the importance of the multiple sensitive volume method. A. Balakrishnan et al. used graph-based neural network methods in the field of reliability modelling and developed a systematic framework for the thorough predictive study of SEU type soft error effects in a particular circuit in [BLG<sup>+</sup>20].

Although there is extensive research on system reliability analysis and prediction, these studies mainly focus on static reliability analysis during the system design phase on certain system levels. However, in environments with dynamic radiation fluxes, such as space, system reliability may

change substantially over a short period of time. Therefore, real-time system reliability monitoring and analysis is very important. Furthermore, system dependability must be handled concurrently at several abstraction levels. Appropriate failure models can be built based on cell libraries for various manufacturing processes at the circuit level. At the architectural level, effective fault resilience measures must be implemented in order to balance dependability, performance, and power. At the application level, algorithm-based mitigation strategies need to be carefully examined while simultaneously leveraging the fault robustness of underlying algorithms. In general, it is a research issue to provide real-time system reliability identification, as well as the common analysis and management of approaches working at distinct levels.

For self-adaptive resilient applications, real-time system reliability analysis is critical for determining optimal adaptive system operation modes to avoid over-/under-protection during run-time. This problem is handled to some extent in [GRH18], where BRAMs integrated in an FPGA are employed as particle detectors, and two pre-designed FPGA bitstreams are used. (i.e., high-performance and high-reliability systems) can be selected for configuration based on the MTUs of the detected faults during run-time. Although this scheme achieves real-time on-chip RAM reliability detection, it does not further analyze the corresponding error rate of the system. In addition, the mode selection of this design is based on the data stream pre-existing in the external memory for real-time configuration, which consumes additional memory resources, and is not suitable for ASIC design. Besides that, real-time system reliability analysis during harsh dynamic environments, such as space, is still an open topic.

### **3.3 Dynamic Adaptation Systems**

The dynamic adaptive system is capable of evaluating and changing its own behavior, whenever the evaluation shows that the current operating mode is not achieving requirements. As introduced in Section 1.2, the focus of this thesis is on the hardware system run-time adaptive approach, which dynamically adjusts to the dynamic radiation environment and system needs in terms of reliability, performance, and power consumption. Such a strategy is appropriate for mission-critical systems that demand real-time operation in dynamic working settings. The present dynamic system designs for Field Programmable Gate Arrays (FPGA)-based and Application Specific Integrated Circuit (ASIC)-based implementations are introduced in the following subsections.

#### **3.3.1 FPGA-based Dynamic Systems**

FPGAs are powerful digital processing devices that designers can reconfigure after manufacturing, and because of this, their main characteristics are complexity and flexibility. FPGA reconfiguration can be partial or full: a full reconfiguration totally replaces the configuration bitstream, whereas a partial reconfiguration just updates a small percentage of the resources. Moreover, due to the flexibility of FPGAs, they have an extremely wide range of applications in many fields [CMA09], such as hardware accelerators, parallel computing, high-end processing



unit, etc. Besides that, the reconfiguration feature of FPGAs supports the implementation of an adaptive hardware system [KVF19], which can be dynamically modified its behaviors at the hardware level to adapt to the working environment, task requirements, etc.

Dynamic FPGA systems are mainly used to meet different application requirements in various scenarios. One typical case for dynamic reconfiguration in FPGA systems is when the high computing needs surpass what the software can supply but proprietary hardware is inflexible. Due to the requirements of flexibility with hardware performance, the dynamic reconfiguration FPGA is very suitable for Software Defined Radio (SDR) applications [DPM<sup>+</sup>07]. In [CJD<sup>+</sup>08], the authors extend the approach to cognitive radio, which is more advanced than SDR, and enables run-time functional modification to operate more efficiently in new environments. Furthermore, rather than having separate basebands, adaptive implementation of the needed baseband characteristics allows them to be altered individually. Another popular adaptive application in FPGAs is devoted to data clustering machine learning modules, such as K-means clustering and Support Vector Machines (SVMs) [HKE12]. Therefore, the target cluster kernels can be customized and hosted on the same FPGA board. Besides that, the concurrent adaptive implementation of multiple kernels can also overcome the requirements of a large number of multiplexed classifiers and also improve the system's overall performance [HBS14]. Due to the rapid development of modern smart automotive, the corresponding software and hardware may need to be updated frequently during the long lifespan of the vehicles. Therefore, the flexibility feature for FPGAs brings benefits for run-time reconfiguration in this field, especially the rapid improvement of driver assistance methods [CSH07]. Furthermore, the authors of [CZM<sup>+</sup>07] suggest a system that employs a processor for control and management, with various image-processing functional units implemented as co-processors and dynamically loaded as needed.

FPGAs' dynamic reconfiguration capabilities can also assist lower system costs by enabling temporal multiplexing of function modules on a tiny chip as needed rather than on a larger one. The comparable applications have been beneficial in audio and video processing applications [TFZ08], where the needed logic operations may be partitioned temporally into smaller tasks. In [RKL10], a scalable advanced video coding deblocking filter architecture based on dynamic reconfiguration is proposed, where the filters can be adapted at run-time to various task user requirements. Furthermore, [BSP<sup>+</sup>09] proposes real-time video processing systems in which several image processing filters are implemented in the same reconfigurable region. Thus, reducing overall power consumption and resource requirements. In [JNK07], a method to achieve network power saving by changing the implementation under different conditions has been proposed, which can monitor the working environment variations and adapt the implementation accordingly.

FPGAs in harsh radiation environments (such as space) can be affected by high-energy particles, especially SRAM-based FPGAs, which are very susceptible to SEU, and can cause configuration memory corruption and severe system corruption [MCE<sup>+</sup>03]. Since the dynamic reconfiguration

in FPGAs provides a secondary path to configuration memory, it has been applied to mitigate the radiation-induced faults on FPGAs. In [BMS07], a partitioning approach for FPGAs is introduced, which intends to isolate SEU faults. Moreover, correct calculations are ensured by applying duplication and comparison, and corresponding regions are reconfigured once errors are detected. Another solution often used in FPGAs to mitigate SEUs [HSW<sup>+</sup>09] is configuration scrubbing, in which configuration data is maintained in radiation-hardened memory and occasionally utilized by the configuration controller to reconfigure components of the FPGA. Besides that, the traditional TMR methods can also be used in FPGAs for advanced fault mitigation. The authors of [SVF<sup>+</sup>13] suggest an enhanced reliability technique based on automotive electronics redundancy with reconfiguration, in which numerous redundant electronic control units are implemented in the FPGA. When the critical condition (i.e., error detection) is triggered, the appropriate region can be changed to recover from the error.

Applications that deal with changing surroundings, as stated above, are great candidates for dynamic FPGA systems. FPGAs may be employed in a variety of dynamic application scenarios and provide additional benefits such as lowering system costs, boosting dependability, and so on since different compute modules can be loaded as needed at run-time. While the majority of these applications are created ad hoc rather than following a generic design, they illustrate the usefulness of dynamic FPGA in a variety of disciplines.

The properties of introduced FPGAs, especially flexibility, also provide the possibility for the implementation of a self-adaptive resilient system. In [JCG<sup>+</sup>12], an FPGA-based self-adaptive reconfigurable fault-tolerant platform is proposed, which enables the system to dynamically adjust the redundancy level and fault mitigation of the system according to the varying radiation incurred at different orbital positions. Furthermore, using existing and publicly available tools, the authors propose a failure rate estimation model and capture the time-varying radiation of different satellite orbits. However, due to the lack of onboard detection methods, this approach may not be able to handle sudden radiation burst situations, such as SPEs. This issue has been addressed in [GRH18], in which an onboard BRAM-based SEU monitor is applied, and a faults' MTTU analyzing method is proposed for the analysis of the real-time solar status. Therefore, the corresponding FPGA board can be configured to high-performance mode during relaxed solar conditions by using no redundancy or the reliability mode to allow better reliability during harsh solar conditions through TMR mitigation. However, this method cannot provide fine-grained solar condition detection and prediction, and only a few coarse-grained system-level redundancies can be reconfigured, which could not meet the diverse needs of the environment and tasks.

### 3.3.2 ASIC-based Dynamic Systems

ASIC stands for an application-specific integrated circuit. Different from the FPGAs, which have the essential features of flexibility and reconfigurability, an ASIC chip is designed and manufactured for one specific application and does not allow users to reprogram or modify it after

it is produced. As a result, ASICs are not usually used for general purposes and the requirements for individual products are regulated, especially in the embedded area. On the other hand, ASICs also have many advantages, such as higher performance, lower energy consumption, less footprint/bill of materials, and thus increased system reliability. Since ASIC-based systems are not particularly flexible, the corresponding dynamic system designs and applications are not as many as FPGA-based systems. Even so, there are still some dynamic systems implemented in ASIC designs, especially for multiprocessing systems.

One of the most prevalent methodologies for dynamic methods is core adaptation. To boost performance and decrease the overhead of thread migration, [SKK15] includes an adaptive morphable core system that can adapt into a multiprocessing system. This work describes an online controllable reconfiguration architecture with four non-monotonic core configuration approaches for varied voltages, frequencies, and resources, with the goal of striking a balance between power efficiency and reliability. In [FNV<sup>+</sup>20], a Functional Units (FUs) level (e.g., multipliers, dividers, ALUs) units configuration method is introduced, in which these FUs can be reconfigured into different operation modes during run-time. As a result, due to the finer granularity, this FU level approach can increase the configuration space and have the options to deal with the issues from the unit itself (e.g., aging). Similarly, a method to extend a processor design with new FUs for catching specific commands and configuring corresponding units with a low configuration latency is present in [FJN19]. The composite core architecture is introduced in [ALU<sup>+</sup>12], which takes the concept of heterogeneity from between different cores to within a signal core and benefits system performance and energy reduction. A dynamic heterogeneous architecture is available in [SKK<sup>+</sup>14], which reconfigures among several core configuration modes with varying voltage and frequency, which may increase overall energy efficiency and perform the optimal configuration mode selection during run-time. [TSF08] introduces a multiprocessor that can dynamically trade-off system reliability, power consumption, and performance. This work makes use of the adaptability of multi-cluster core computers, which may allocate different cores to different tasks.

In the Very Long Instruction Word (VLIW) architecture, where the available slots may execute one long instruction that is made up of numerous small instructions in parallel, the dynamic hardware redundancy can be employed to improve system performance and reliability. In [MSC10], hardware- and software-based cooperative detection technique is proposed, which can make the data paths of scheduled superscalar processors fault-tolerant. The fundamental aim is to complete the identical work in both slots as long as the outcomes are consistent. In the case of a mismatch, a third-time slot is provided dynamically to form a TMR system. When high performance is necessary, the program uses all available slots in a non-fault-tolerant manner without repeating operations. Similarly, to alleviate soft mistakes, [PKS17] proposes an adaptive fault-tolerance system in which a hardware mechanism is provided to reproduce and schedule the instructions at run-time, thereby utilizing idle functional units to execute repeated instructions in a configurable VLIW processor.

It can be seen from the above introduction that whether it is a multiprocessing system or a VLSI system, there are many ASIC-based dynamic applications that utilize the redundant resources of the existing on-chip hardware to achieve the trade-off between performance, power consumption, and reliability during run-time. However, due to the limited flexibility, the dynamic modes that can be implemented are relatively limited, generally limited to coarse-grained dynamic adjustment of specific predefined modules.

It is not enough to be able to adjust the operating modes for a self-adaptive resilient system, where the system should also be aware of the environment and functioning state during run-time. These requirements have been addressed to a certain extent in [ASII14], where an architecture framework for dynamically adaptable multiprocessing systems is presented, which intends to achieve the dynamic trade-off between core aging, system reliability, power consumption, and performance. In this work, several embedded aging monitors are applied to detect the individual core ages, and dynamic core-level NMR methods are applied to adapt to the different system requirements. Besides that, the youngest-first round-robin core-level pattern approach is used to greatly increase the system's lifetime. Although this design greatly improves the flexibility of ASIC-based systems and can sense the core status and make corresponding mode adjustments to a certain extent, there are still some points that can be improved. First of all, in the above work, only the status of the system can be identified through sensors, but changes in the external environment cannot be identified, and thus cannot deploy the necessary modes before the onset of the critical situations. Besides that, a limited number of core-level redundancy methods can be used, and fine-grained tuning under dynamic conditions and requirements cannot be achieved.

### **3.4 Summary of Open Issues Addressed in this Thesis**

This section details the crucial outstanding questions in relation to the relevant research stated in the preceding parts, particularly on the subject of self-adaptive resilient systems for space applications, which is the major emphasis of this thesis:

1. Particle detector: the current particle detectors used for radiation-induced fault monitoring lack the critical properties needed for a self-adaptive resilient system. The main disadvantage is the inability to accurate detection, i.e., the need to implement detection of transient and permanent faults and distinguish between multiple error types. Moreover, the detector needs to have a sufficient sensing area and be easily implementable in the embedded system. As a result, it is important to investigate a novel onboard detector for real-time radiation intensity monitoring, capable of supporting the related low-cost in-flight measurement for the embedded system in space applications.
2. Space weather and reliability analysis: the main drawback of the existing methods is that they cannot provide a fine-grained in-flight onboard prediction for embedded systems. In addition, there is no method to combine the system reliability prediction and the corresponding radiation particle intensity. Therefore, this imposes the need for low-cost

onboard in-flight target modules (i.e., SRAM blocks in this thesis) for real-time fault rate prediction. According to the predicted fault rate, it is possible to perform real-time space weather prediction and forecast extreme solar events earlier.

3. Dynamic adaptive system: due to the pre-designed hardware architecture, the dynamic system generally has limited patterns formed, especially for ASIC-based systems. Thus, it could not be sufficient enough for fine-grained operation mode selections for various working conditions and task requirements. Besides that, the mode selection methods should be optimized for the trade-off between system reliability, power consumption, and performance. Therefore, it is essential to determine the methods for fine-grained operation mode selection, as well as methods for analyzing the environment and system status to determine the optimal operation modes under mix-critical applications in harsh radiation environments.

Each of the aforementioned outstanding topics is addressed in this thesis. As a result, a method for developing self-adaptive resilient processing systems, particularly multiprocessing systems, can be developed. The target system intends to detect and forecast real-time working environments, analyze system status, construct dynamic operation modes, and select optimal operation modes during run-time. The proposed solutions are discussed in depth in the following chapters.

## Chapter 4

# SRAM-based Radiation Monitor

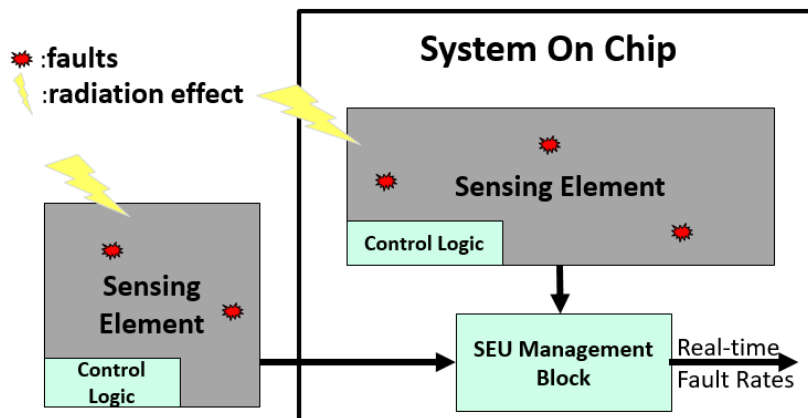
To enable fault resilience in an autonomous manner for space applications, real-time in-flight detection of space radiation conditions is necessary. The intensity of cosmic radiation can change by five orders of magnitude within hours or days, as detailed in Section 2.1, increasing the chance of soft errors in space-borne electronic systems by several orders of magnitude. Therefore, monitoring the real-time space environment is essential to ensure the timely detection of high radiation levels and efficient protection of radiation-sensitive circuits in space-born applications. This chapter proposes and discusses a non-standalone embedded SRAM-based Single Event Upset (SEU) monitor, which combines the SRAM SEU monitoring and data storage functions in the same on-chip Static Random Access Memory (SRAM) module, with negligible cost and overheads, compared to traditional stand-alone SEU monitors. Furthermore, the proposed monitor also enables the correction or detection of multiple-bit upsets and permanent faults in each memory word. To identify run-time radiation conditions, the proposed monitor is intended to be integrated into a self-adaptive resilient processing system for radiation environment detection.

The chapter is divided into four sections. Section 4.1 presents the concept of the proposed radiation monitor. The architecture and operation of the proposed SRAM-based SEU monitor are described in Section 4.2. Section 4.3 introduces the implementation of the proposed design with two different cases. Section 4.4 provides an analysis of the suggested monitor design. The summary is outlined in Section 4.5.

### 4.1 Concept

Figure 4.1 presents the general concept of the proposed monitor for in-flight radiation-induced faults detection. As introduced in Section 3.1.6, for self-adaptive resilient processing systems, the particle monitor needs to be placed close to the target system in order to detect the radiation conditions to which the target system is exposed during the mission. Therefore, existing onboard components (i.e. embedded SRAM in this thesis) in ICs can be reused as sensing elements. The radiation particles (i.e. heavy-ion and proton) could cause events in the target sensing elements,

such as SEU, multiple-bit upsets and permanent faults. With a dedicated control logic, the proposed radiation monitor can realize accurate fault detection, and error type classification and prevent over-counting without affecting the basic functionality of the sensing elements. Moreover, one SEU management block gathers the detected faults information during each pre-defined time period (e.g. one hour). This management block also provides a data interface for other blocks to access the information it collects.



**Figure 4.1:** General concept of the proposed monitor with the sensing elements inside an embedded system. The green blocks represent the main focus of this dissertation.

### Sensing Element

As shown in Figure 4.1, the on/off-chip sensing element in the target system is intended to sense the effects of radiation particle strikes, especially SEUs. To ensure correct radiation-induced fault detection, the proposed monitor, especially the sensing element, should obey the general good sensor design rules [CRE<sup>+</sup>19]:

- Sensitive to the measured property
- Insensitive to any other property likely to be encountered in its application
- Does not influence the measured property

In order to ensure that the sensing element and the target system are affected by the same radiation environment, the existing on-chip components can be reused as the sensing element. In general, radiation-induced faults in all on-chip components have both advantages and disadvantages. Table 4.1 shows the suitability of the most commonly used basic ASIC components as sensing elements. The SEE regards the commonly occurring SEE type of the target element. Since the RAM can also be integrated externally, thus, the potential for scalability is much higher than for purely embedded fixed on-chip elements. The distribution refers to the area coverage across the chip [ASII14]. Note that only the most commonly used basic ASIC elements are qualitatively compared in the table. Moreover, the radiation sensitivity according to the critical

charge is analyzed in detail in Subsection 4.4. As detailed in Section 3.1.5, RAMs are sensitive to radiation particles with large sensing areas. Moreover, since DRAM contains an external memory refresh circuit that could prevent fault accumulation, SRAM has a higher radiation sensitivity. By considering the above analysis and good sensor design rules, SRAM is the best option as the on-chip sensing element for embedded applications:

- High sensitivity to radiation particles
- Sufficient scalability and distribution, easy to expand
- Independent memory blocks, simple to access and locate the faults

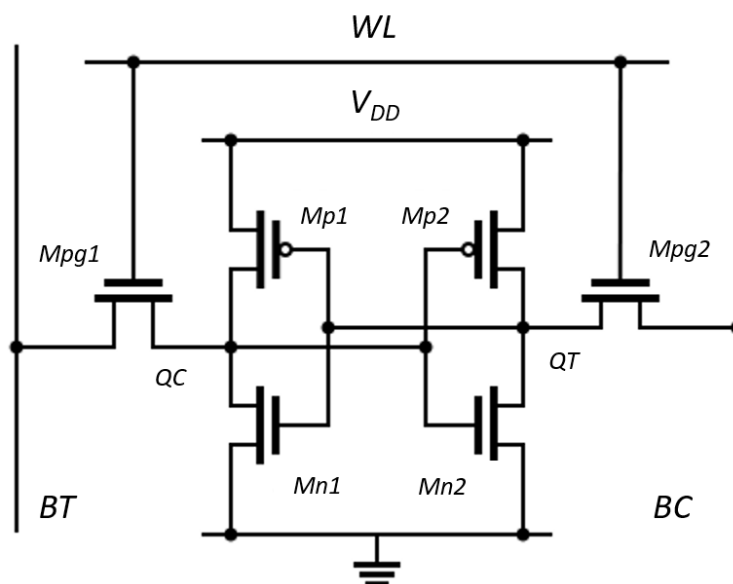
**Table 4.1:** The suitability of ASIC on-chip components as sensing elements. The SRAM is chosen as the sensing element in this thesis.

Element	Radiation Sensitivity	SEE	Scalability	Distribution
SRAM	Very High	SEU	High	High
DRAM	High	SEU	High	High
Combinational Gates	Low ~ High	SET	Low	Middle
Flip-Flop	Low	SEU	Low	Low

## SRAM

SRAM is a form of volatile memory that stores bits via latching circuitry. SRAM memories are often manufactured using CMOS bulk or SOI technology. The most common memory cell layout consists of six MOSFETs, which are insulated-gate field-effect transistors. Such architecture, six-transistor (6T) SRAM, is illustrated in Figure 4.2. Other SRAM architectures, in addition to the 6T SRAM, employ 4, 8, 10 (4T, 8T, 10T SRAM) or more transistors per bit. The memory element of the cell is a latch implemented by two cross-coupled inverters (transistors Mp1/Mn1 and Mp2/Mn2). The other two NMOS transistors (Mpg1 and Mpg2) are necessary for controlling the read and write operations. The most radiation-sensitive nodes within the SRAM cell are the QC and QT nodes. When a particle can create enough charge, either directly or indirectly (through ion recoils), to be gathered at these nodes, the parasitic current generated would compel the node in the OFF state to discharge, which could result in the bit cell's entire state shift. Compared to other logic gates, SRAM cells usually exhibit higher sensitivity to radiation and are thus suitable as radiation monitors. In general, the overall sensitivity of the SRAM module is determined by the number of cells, i.e., by the total memory capacity.





**Figure 4.2:** Architecture of a six transistor SRAM cell.

The SRAM has three basic operations: retention, read operation and write operation. In the retention mode, the access transistors are turned off, and the cell is isolated. Both BT and BC are pre-charged at VDD. The information is stored in the latch created by the two-inverter loop. BT and BC are initially pre-charged at VDD and then left floating during a read operation, allowing them to function as capacitors with the same potential. By setting the WL to VDD, the access transistors are turned on. Two cell nodes are in opposing states, one at 0V and one at VDD. Because the node and the BC are at the same potential, the BT connection to the node at 0V can experience a little discharge, whilst the other would not. Which BT has been significantly discharged can be detected using a differential sensing amplifier. This information, in turn, can supply the logic state stored in the cell in the output. Finally, depending on the bit to be saved, the BTs can be changed to 0V or VDD to modify the logic information stored within the memory cell. The access transistors are then switched on. As a result, if the write driver has a high drive strength, the inverter loop's state can be changed.

Several more electrical components are required to activate certain tasks, select memory cells, read stored information, send or receive data, synchronize, and so on in order to conduct the aforesaid actions on SRAM. Furthermore, in addition to memory cells, the radiation environment could also have an impact on the remaining electrical circuits. As a result, a proper policy for protecting the remaining electrical circuits is also required. To mitigate these effects, the control logic can be protected with radiation-hardening-by-design techniques, such as the combination of the TMR method. Moreover, the TMR flip-flops [VPK15] can also be used in the SRAM control logic part to enhance the robustness against radiation particles.

### SRAM with radiation-induced characterization

For space-borne applications, the radiation-induced faults in SRAM are generally caused by protons (mainly indirect ionization) and heavy-ions (direct ionization), and other particles can be neglected [EPE11]. As detailed in Section 2.2, the basic concepts are different for heavy-ion and proton-induced upsets. A single heavy ion could pass through the silicon of a memory cell and deposit a track of ions, thus, resulting in an SEE. On the other hand, protons can mainly initiate indirect ionization with the particle recoil in the silicon device, and these reactions can produce heavy ions that can deposit enough energy to cause SEEs.

To undertake a quantitative examination of the radiation particle sensitivity of the memory device, proton and heavy-ion radiation tests on the target SRAM are required to acquire the corresponding cross-section data. The cross-section,  $\sigma$ , is a statistical representation of the sensitive node or sensitive volume for a specific SEE, represented as a relative area in  $\text{cm}^2$ . An electrical component's SEE cross-section can estimate the possibility that a particle of particular energy generates a SEE. A particle must deposit a critical charge in the sensitive volume to form a SEE. The critical charge,  $Q_c$ , means the minimum amount of charge to cause an upset in the cell. For proton testing, the cross-section is usually calculated as a function of the proton's energy (in MeV). In the case of heavy ions, it is calculated as a function of the ionizing energy transferred by the ion to the material per unit length divided by the substance's density. This quantity is called LET (Linear Energy Transfer) in  $\text{MeV}\cdot\text{cm}^2/\text{mg}$ . During the radiation testing, it is feasible to assess the memory's sensitivity at a macroscopic level by calculating its cross-section using the following equation by recording and counting the number of events caused by a single particle on a specific component:

$$\sigma = \frac{N_{event}}{F * \#bits} \left( \text{in } \frac{\text{cm}^2}{\text{bits}} \right) \quad (4.1)$$

where  $N_{event}$  is the number of events generated, and  $F$  is the particle flux or the number of particles per  $\text{cm}^2$ ,  $\#bits$  is the amount of memory in bits. The aforementioned equation allows the sensitivity to be compared at the bit level, making it easier to compare different memory cross-sections of varying widths or bit capacities.

$$\sigma(x) = \begin{cases} \sigma_{sat} (1 - e^{-\left(\frac{x-x_0}{w}\right)^s}), & \text{if } x > x_0 \\ 0 & , \text{ if } x \leq x_0 \end{cases} \quad (4.2)$$

As it is impossible to test all and every LETs (for heavy-ion) or energy (for proton) the cross-sections, only a limited number of LETs and energies cross-sections can be obtained from the radiation tests. Therefore, the interpolated and extrapolated method is necessary to estimate the cross-section event for the other LETs and energies. The Weibull distribution function, defined in

Eq. (4.2), is a widely used method to model and fit the experimental results from radiation testing at particle accelerators. The Weibull function is generally characterized by four parameters: the on-set parameter ( $x_0$ ), the saturation cross-section ( $\sigma_{sat}$ ), the width parameter ( $W$ ), and the dimensionless exponent ( $S$ ). For proton cross-section, the corresponding equation can be represented as:

$$\sigma(E) = \sigma_{sat} \left(1 - e^{-\left(\frac{E-E_{th}}{w}\right)^s}\right), \text{ if } E > E_{th} \quad (4.3)$$

where  $E_{th}$  is the energy threshold (on-set) of proton in MeV,  $W$  is the scale parameter in MeV,  $\sigma_{sat}$  is the saturated cross-section in  $\text{cm}^2$ . For heavy-ions, the Weibull distribution function of the cross-section is shown in the following equation:

$$\sigma(LET) = \sigma_{sat} \left(1 - e^{-\left(\frac{LET-LET_{th}}{w}\right)^s}\right), \text{ if } LET > LET_{th} \quad (4.4)$$

Where  $LET_{th}$  is the  $LET$  threshold of heavy-ions in  $\text{MeVcm}^2\text{mg}^{-1}$ ,  $W$  is given in  $\text{MeVcm}^2\text{mg}^{-1}$ .

A Cypress COTS SRAM manufactured in 65 nm bulk CMOS technology is chosen as the target SRAM for analysis in this thesis as a case study. In [VGU17], a series of heavy ion and proton radiation tests have been carried out on the target SRAM, and the experimental data have been fitted to the Weibull function, defined by Eq. (4.3) and (4.4). There are four types of SRAM cross-section values obtained from radiation tests [VGU17]: bit or event cross-section with static or dynamic operating mode. The event cross-section is calculated by counting the number of events instead of the number of bit flips. Based on how many bits are flipped, bit cross-sections represent the memory's SEE sensitivity in usage. Viewing event cross-sections can help determine the kind of event triggered by a single particle, which can be useful in deciding the appropriate ECC method. Furthermore, the cross-section is influenced not just by radiation beam characteristics but also by how the memory is activated during irradiation. Thus, the memory needs to be tested by using two types of testing methods: static (i.e. idle state) and dynamic (i.e. read/write). In the static mode, the memory stores the constant information and works in the idle status during the exposure. For the dynamic test, where different algorithms can access the memory through reading and writing operations, e.g., the March C-algorithm. Furthermore, the dynamic test mode allows for stress on memory cells as well as any peripheral regulatory circuitry that may be radiation sensitive. According to [GTS<sup>+</sup>14], the static mode is the most stable method for collecting SEUs instead of dynamic mode testing for SRAMs, in which certain disturbances may occur (for example, Single Event Functional Interrupts) during dynamic tests. Generally, compared to the static mode, the SRAM fault rates are usually lower during the dynamic mode because the write-in procedure may erase the flipped states, thus, the information may be lost.

The corresponding Weibull fitting parameters can be estimated based on the bit or event cross-section with static or dynamic operating mode cross-section data for the target 65 nm SRAM. The Weibull fitting parameters for the target SRAM heavy-ion and proton cross-sections are shown in Tables 4.2 and 4.3, respectively [VGU17].

**Table 4.2:** Weibull fitting parameters for the target SRAM heavy-ion bit or event cross-section with static or dynamic operating mode.

Heavy-ion	Bit cross-section Weibull parameters				Event cross-section Weibull parameters			
	$\sigma_{sat}$	$LET_{th}$	W	S	$\sigma_{sat}$	$LET_{th}$	W	S
<b>Static</b>	$7.90*10^{-8}$	1.04	23.57	1.43	$6.07*10^{-9}$	1.04	4.20	0.6
<b>Dynamic</b>	$6.44*10^{-8}$	0.97	32.37	1.42	$6.26*10^{-9}$	0.97	8.9	0.7

**Table 4.3:** Weibull fitting parameters for the target SRAM proton bit or event cross-section with static or dynamic operating mode.

Proton	Bit cross-section Weibull parameters				Event cross-section Weibull parameters			
	$\sigma_{sat}$	$LET_{th}$	W	S	$\sigma_{sat}$	$LET_{th}$	W	S
<b>Static</b>	$1.97*10^{-13}$	3.00	5.02	0.95	$1.79*10^{-13}$	3.00	3.00	0.50
<b>Dynamic</b>	$7.43*10^{-14}$	3.00	8.85	0.57	$5.52*10^{-14}$	3.00	3.59	0.30

After calculating cross-section of the sensing element, i.e. SRAMs in this thesis, the corresponding upset rate in the space environment of the sensing element can be estimated. The corresponding methodologies can be organized into two general categories [JCP96]. One of the methods is to describe a macroscopic view of the entire chip with responses from charged particle impacts. The other method is to use a microscopic view of individual circuit cells, which is also applied in this thesis. In [EPE11], the proton and heavy-ion induced upset rate calculation method is described. In this thesis, the Cosmic Ray Effects on Micro-electronics Model 1996 (CREME 96) [AJT97] [CRE96] is used for the sensing element in-orbit upset rate estimation. The CREME 96 is one of the most widely used suites for evaluating the in-orbit SEU rate. Moreover, the CREME96 SPE models (i.e., Peak Five Minutes (P5M), Worst Week (WW), and Worst Day (WD) modules) are based on the corresponding average flux data of the October 1989 event, which is one of the most significant events in past decades. In the CREME96 HUP model, which is used for the direct ionization-induced SEE rate calculation, the heavy-ion cross-section must be specified in bits but not for the device. This requirement is that CREME96 SEE-calculations are based on RPP (Rectangular Parallelepiped Parallelogram) model [JCP80]. A vital ingredient of the RPP model is the distribution of path lengths through the sensitive volume of the bit. Therefore, only the proton and heavy-ion bit cross-sections in Tables 4.2 and 4.3 have been considered for the target SRAM in-orbit upset rate estimation. Moreover, a shielding parameter of 100 mils is assumed during the estimation, which is the conventional equivalent shielding thickness for spacecraft [CRE96]. In-orbit estimates of proton-induced and heavy-ion-induced SEU rates on the target SRAM in static and dynamic modes for five CREME96 space radiation particle models are provided in tables 4.4 and 4.5, respectively. In all models, it can be clearly found that the static mode has a higher SEU

rate than the dynamic mode. On the other hand, in dynamic mode, the SEU rate is lower because the write-in procedure may erase the flipped states, and thus the fault information may be lost. Therefore, although our embedded memory could, in practice, operate in dynamic mode, at the system level, it would be necessary to ensure tight synchronization between writing in the memory and counting the bit flips. In other words, before new data is stored in the memory, it would be necessary to check its contents to detect possible bit flips. Thus, it could be possible to regard the memory operation as static between the two readout sessions. This thesis chooses parameters related to static mode tests and bit cross-sections for SRAM SEU rate estimation based on the above analysis. Therefore, the bit cross-section with static mode data of the target 65 nm SRAM in Tables 4.2 and 4.3 are chosen as the example for the following space radiation particle-induced faults analysis.

**Table 4.4:** Heavy-ion induced SEU rate with the static and dynamic Weibull fitting parameters in different CREME96 space radiation particle models.

CREME 96 Models	Heavy-ion induced SEU rate (upsets/bit/day)	
	Static	Dynamic
Solar Minimum	$3.25 \times 10^{-7}$	$1.81 \times 10^{-7}$
Solar Maximum	$8.84 \times 10^{-8}$	$4.89 \times 10^{-8}$
Worst Week	$6.97 \times 10^{-5}$	$3.91 \times 10^{-5}$
Worst Day	$3.21 \times 10^{-4}$	$1.78 \times 10^{-4}$
Peak 5 Minutes	$1.19 \times 10^{-3}$	$6.62 \times 10^{-4}$

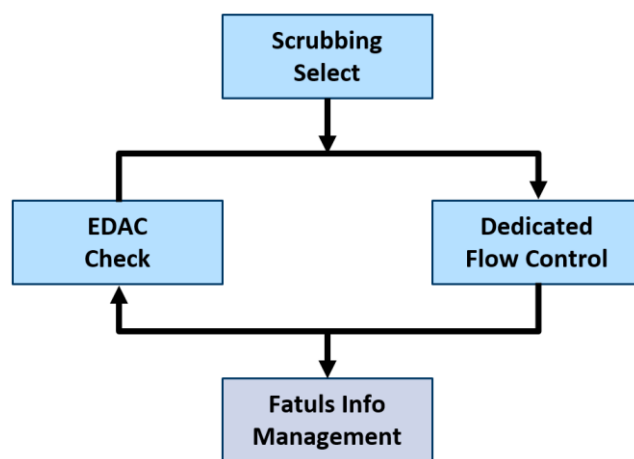
**Table 4.5:** Proton induced SEU rate with the static and dynamic Weibull fitting parameters in different CREME96 space radiation particle models

CREME 96 Models	Proton induced SEU rate (upsets/bit/day)	
	Static	Dynamic
Solar Minimum	$7.57 \times 10^{-8}$	$2.85 \times 10^{-8}$
Solar Maximum	$2.68 \times 10^{-8}$	$1.01 \times 10^{-8}$
Worst Week	$1.67 \times 10^{-4}$	$5.44 \times 10^{-5}$
Worst Day	$7.41 \times 10^{-4}$	$2.39 \times 10^{-4}$
Peak 5 Minutes	$2.74 \times 10^{-3}$	$8.82 \times 10^{-4}$

## 4.2 Single-Event Upset Monitor Design

In this section, the design method of the proposed SEU monitor is introduced in detail. As shown in Figure 4.3, the proposed monitor mainly consists of 4 major modules: a scrubbing module for memory word selection; an Error Detection And Correction (EDAC) module for fault checking in each word; a dedicated fault detection flow with an Over-counting Detection Register File (ODRF) to ensure the accurate count of all bits upsets that are occurring, and distinguish the SEUs and MCUs as well as permanent faults in each memory word; one fault management module to manage

the current and historical fault info from all sensing elements. An embedded non-standalone SRAM-based SEU monitor is offered as an alternative to the traditional stand-alone RAM-based particle detectors described in Subsection 3.1.6 [CAS<sup>+</sup>19]. The usual on-chip SRAM memory can be used as a particle detector in addition to its typical data storage function. The detection technique is similar to the stand-alone SRAM detectors explained in Subsection 3.1.6, in that it counts the detected faults in SRAM cells and determines the corresponding particle flux. Furthermore, the target SRAM can be used for in-flight SEU measurement while maintaining its purpose as a data storage device. As a result, the suggested monitor can be readily and cheaply installed in embedded devices. Aside from that, an important characteristic of the suggested method is the capacity to identify and discriminate MCUs and permanent defects from MCUs in SRAM cells, which is a unique feature of stand-alone particle monitors to the best of my knowledge. The aforesaid characteristics are critical for sustaining accurate SEU measurements in long-term missions when permanent flaws emerge owing to slow device wear-out and each particle can cause several upsets in memory cells in cutting-edge technology. The details of the introduced four significant modules in the proposed monitor will be detailed in the following subsections.



**Figure 4.3:** General workflow of the proposed monitor with four main modules.

#### 4.2.1 Scrubbing

The scrubbing can verify and correct information stored in the memory [PIM19]. This method is generally used for avoiding fault accumulation and mitigating data corruption in memory banks. There are two main traditional types of scrubbing: read back and blind scrubbing. The read-back scrubbing approach can first check if the memory word is corrupted and then evaluate redundant information of the corresponding data word to correct the error information. In contrast to the read-back method, blind scrubbing can write the entire memory without checking the corruptions of memory words. Thus, for blind scrubbing, a golden reference data of the memory is required. The procedure of overwriting the bits with their good values, known colloquially as "data scrubbing,"

is carried out at a time frequency appropriate for the projected rate of radiation-induced errors and in a way that does not jeopardize the IC's performance. Scrubbing avoids the buildup of damaged bits in memory banks that store accessible configuration data or any other data. In addition, periodic scrubbing can also minimize the accumulation of transient faults and implicitly the probability of multiple-bit errors in memory banks [SSM00].

This thesis applies the read-back scrubbing method in the proposed monitor for periodically checking the sensing elements, i.e. SRAM blocks. The introduced scrubbing module is intended to collaborate with the EDAC module for the period of memory word checking and scrubbing. As opposed to the traditional blind scrubbing method, the "golden value" is not required and is replaced by the EDAC module, which can significantly reduce the cost. As shown in Figure 4.3, during the SEU checking period, the scrubbing module would sequentially select a few data of the sensing elements, i.e. one memory word in this approach, and send the corresponding data to the EDAC and dedicated detection flow modules for checking. Moreover, after the detection and correction processes from EDAC and the detection flow modules, the corrected memory word can also be written back to the same address. Moreover, according to [PIM19], the scrubbing method can also limit multiple errors by accumulating single errors over time, which is essential for the accurate faults count of faults in the proposed monitor.

#### 4.2.2 EDAC

ECC or Forward Error Correction (FEC) are algorithms that identify and/or rectify data mistakes by adding redundant or parity data to the original data. The term EDAC is used when mistakes are recognized and repaired. EDAC is an information redundancy method that belongs to the linear coding family [SLJ12]. As the result of adding redundant or parity bits, the memory word can be retrieved even if multiple faults (up to the capabilities of the code being used) occur during transmission or storage. To minimize the system's error rate, error-correcting codes are widely employed in lower-layer communication and for dependable storage in media like CDs, hard drives, and RAMs. Moreover, EDAC codes are widely used to increase computer memory reliability in space applications. For example, additional parity bits in each memory word can be calculated during the word writing process and used for corruption bit detection/correction during the reading process. Each coding scheme has unique fault detection and fault repair features, but they all have an influence on the system by adding an area overhead to store the redundant data and a time cost to calculate these data and check the original data for consistency. The EDAC method is a practical approach to improving memory subsystems' reliability and yield since they can be deployed in memory blocks with limited design overhead. As an example, Table 4.6 shows the experimental results of the target 65 nm SRAM (with 16 Mbit) with Fe heavy-ion irradiation to assess the memory's responsiveness with and without a built-in internal ECC [VGU17]. In static mode testing, the majority of the bits can be rectified, and just a few faults remain. Because the ECC technique employed in the experiments is based on parity bits, the non-corrected bits are

almost MBUs. Furthermore, because faults do not accumulate in dynamic mode, the ECC may repair almost all of the bits in the corresponding tests. As a result of the testing results, the ECC provides for a one to two orders of magnitude reduction in the cross-section of the target SRAM.

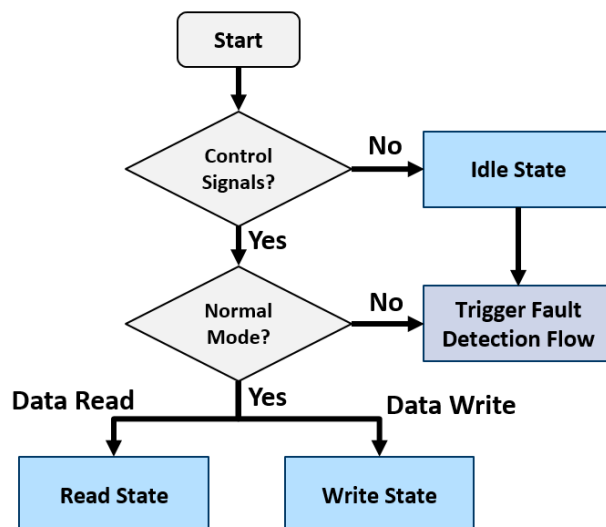
**Table 4.6:** Fe ion ( $LET = 19 \text{ Mev.cm}^2/\text{mg}$ ) radiation test results of a 16 Mbit SRAM with and without ECC for static and dynamic tests [VGU17].

Type of Test		ECC OFF		ECC ON	
		Total number of SEUs	Cross section (cm <sup>2</sup> /bit)	Total number of SEUs	Cross section (cm <sup>2</sup> /bit)
Static	<<0000>>	5940	$3.22 \cdot 10^{-8}$	51	$3.01 \cdot 10^{-10}$
Static	<<1111>>	5216	$3.02 \cdot 10^{-8}$	192	$1.13 \cdot 10^{-9}$
Static	<<0101>>	6076	$3.32 \cdot 10^{-8}$	26	$1.52 \cdot 10^{-10}$
Dynamic	C-normal	2542	$1.47 \cdot 10^{-8}$	1	$5.84 \cdot 10^{-12}$

Each EDAC algorithm has its specific strengths and limitations, and each application has its requirements. In this thesis, the general requirement for the EDAC algorithm is to detect various fault types (SEU or MBU) in the sensing element at a low cost. Thus, an easy-to-deploy flexible fault detection method is required. There are plenty of EDAC codes with different fault correction or detection capabilities [CRF22]. For example, one of the most famous methods is Hamming codes [RWH50], which Richard W. Hamming introduced in 1950. Section 4.4.1 describes the introduction of the other regularly used EDAC codes.

In this thesis, all general EDAC algorithms can be coordinated with the proposed fault detection flow and scrubbing module for real-time SEU rate detection. After the memory word selection from the scrubbing module, the detection flow can control the EDAC module to perform the necessary error checking, confirming over-counting preventing processes, etc. For example, the EDAC codes in this thesis are mainly referred to as HSIAO SEC-DED [MYH70], which is commonly applied for SRAM protection due to its favourable recovery capacity from multiple errors. Hsiao codes are optimized Hamming codes with an odd-weight column. Each codeword in a  $(n, k)$  linear Hsiao code contains  $n$  bits, with  $k$  data bits and  $n-k$  check bits. A parity check matrix,  $H$ , with  $n-k$  rows, one for each check bit, and  $n$  columns, one for each bit in the codeword, can be used to represent the code. To make the code SEC-DED, the  $H$ -matrix must be built in such a way that the minimum distance between any codewords is 4. Since the proposed monitor is intended to be integrated into the embedded systems, which are usually 32-bit in width, as an example, the HSIAO (39, 32) SEC-DED code is selected and implemented in this thesis.



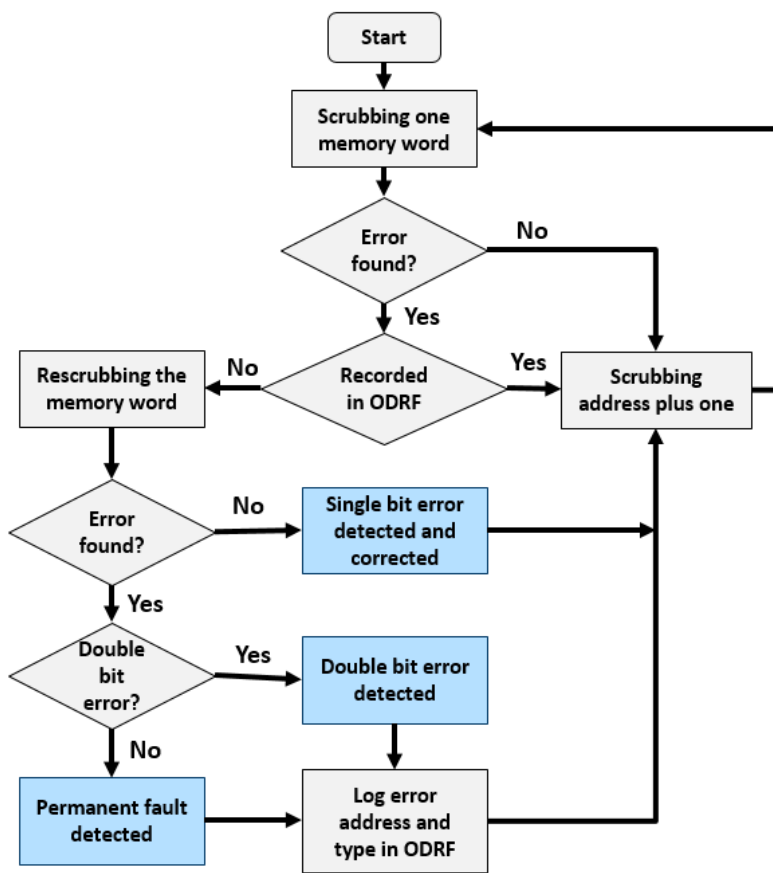


**Figure 4.4:** Sensing elements working mode decision process.

### 4.2.3 Detection flow

The central part of the proposed SEU monitor is the dedicated detection flow, which works as the control logic for the sensing element, EDAC, and scrubbing modules. As shown in Figure 4.4, sensing elements (i.e. SRAM banks) can work in three states: read, write and idle. After receiving the read or write requests from the host system, the memory block can work as a regular storage element and load or store data, respectively. Moreover, the EDAC module can encode and decode corresponding data during the standard data access process. There are two ways to trigger target fault detection procedures: active and passive. After receiving the start detection control signal from the host system or a watchdog timer, the fault detection flow can be passively triggered and check all the memory words in a sequence. In addition, the detection flow can also be automatically actively triggered during the idle state.

After triggering the fault sensing element, a dedicated fault detection flow works for the correction/detection of potential faults in each memory word, as shown in Figure 4.5. This flow is simply piggybacked on EDAC (i.e. HSIAO SEC-DED) and scrubbing mechanisms. The scrubbing module can periodically read memory words, and automatically increment the next scrubbing address, after completing the current scrubbing cycle. The EDAC module is used to check the corresponding selected memory word and provide necessary error info. Moreover, the ODRF has been designed to log erroneous addresses in order to avoid over-counting of the uncorrected errors in the memory word, such as the MBUs and permanent faults. In order to trigger the fault detection flow, the scrubbing module has to be in operation. Upon receipt of the detection start signal, the scrubbing procedure starts from the first address of the memory banks to check each memory word

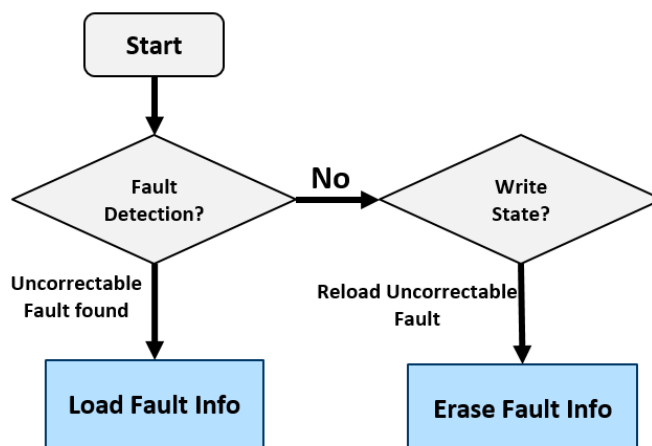


**Figure 4.5:** Dedicated faults detection flowchart of the proposed SEU monitor with the HSIAO SEC-DED EDAC code, scrubbing and Over-counting Detection Register File (ODRF).

with its corresponding parity bits, i.e. 32-bit data with its 7-bit HSIAO (39, 32) syndrome. If no error is detected in the current memory word, or the corresponding address has already been logged in the ODRF, the detection flow can proceed to the next address. On the other hand, if a new error is found, the current memory word needs to be re-scrubbed immediately. If the EDAC module cannot find the previous error again in the new scrubbing round, the corresponding error has been corrected, and it can confirm that one single-bit error has been detected and corrected. On the other hand, during re-scrubbing round, if the EDAC module can still find the error, the double-bit error or the permanent fault can be detected and the 'error type' info from the module can be used for error classification. If the EDAC module shows that a single-bit error is again found in the re-scrubbing round, the corresponding 'single-bit error' should be marked as a permanent fault. Furthermore, the error address of the double/permanent fault should be logged in the ODRF. Otherwise, the duplicate counting of the same double-bit errors and permanent faults cannot be avoided, and the corresponding counters could quickly overflow. In these ways, the proposed monitor can provide accurate in-flight fault counting capability during run-time, which is essential for the following solar condition and SEU rate prediction (detailed in Chapter 5). In addition, three

error counters are used to count the single-bit error, double-bit error, and permanent fault, respectively.

The operational mode flow of the ODRF is shown in Figure 4.6. The ODRF is composed of a set of registers used to log uncorrectable fault addresses to avoid over-counting the same error multiple times after triggering the fault detection flow. In addition, during the writing state of the sensor element, the written address can be compared with the existing load address in the ODRF. If the corresponding address is rewritten, the corresponding fault info can be erased.

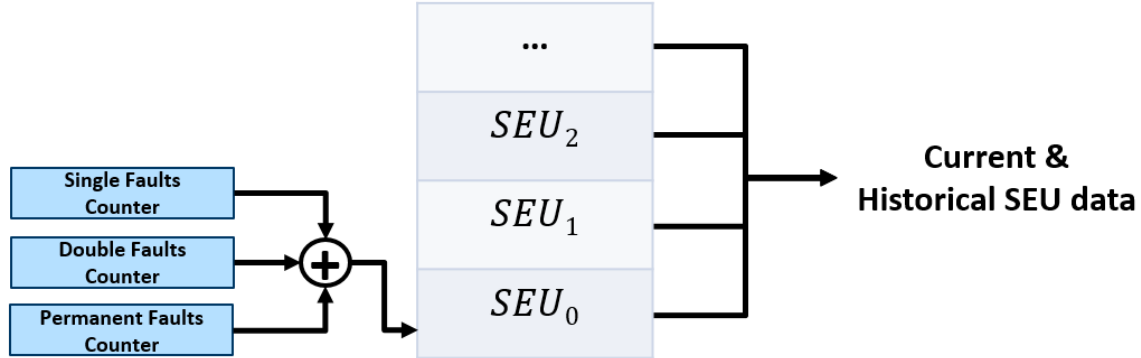


**Figure 4.6:** Workflow of the Over-counting Detection Register File (ODRF).

As shown in Figure 4.5, different types of detected errors can be counted by corresponding counters. After each detection period finishes, the sum of these counter data can be loaded into the SEU management block, which is detailed in Subsection 4.2.3. If there is a counter overflow, the counting starts again from zero, but the corresponding overflow bit is also set in the status register. According to SEU rate analysis of the target SRAM from existing space ions during solar cycle 24 (2008-2019), the peak hourly SEU rate of the target SRAM is  $1.32 * 10^{-3}$  upsets/bit/day. Therefore, if the sensing element is considerable, such as 2 Gbit, the expected hourly SEU count is 118122. Thus, with periodic scrubbing and rewriting as well as resetting, the overflow is not expected for a 32-bit error counter, even during significant solar events.

In addition, in order to ensure the overall detection of the potential SEUs and avoid the accumulation of soft errors, regular scrubbing of all memory words (e.g. once an hour) is necessary. In addition, memory banks cannot be accessed by the user during the scrubbing process, and the system needs to stall and wait for the end of the detection flow. As the example in [CAS<sup>+</sup>19], the SEU checking the time of the detection flow for the entire 20-Mbit SRAM is 42 ms when the working frequency is 50MHz.

#### 4.2.4 Fault Management Block



**Figure 4.7:** Block diagram of the fault management unit.

The SEU management unit gathers detected single and double as well as permanent faults from the internal and external memories in each scrubbing period. The block diagram of the SEU management unit is shown in Figure 4.7. The counted error data is accumulated from the corresponding error counters, as explained in Section 4.2.3, and the total error data is denoted as the period SEU (such as hourly SEU) and can be sequentially stored in the internal memory stack for analysis and statistics. Assuming that the scrubbing period of the memory cells is one hour, the  $SEU_n$  in Figure 4.7 stands for the detected hourly SEU number from the scrubbing of all memory cells  $n$  hours ago, and  $SEU_0$  denotes the latest total counted SEU number. After completing one scrubbing cycle, new hourly SEU data in the memory banks is available and can be pushed into the management unit stack. The output signal of the management unit is the latest detected hourly SEU data, which can directly trigger the optimal mode selection in the resilient system (detailed in Chapter 6), or the combination of historical hourly SEUs for the solar condition and future SEU rate prediction (detailed in Chapter 5).

The SEU monitor utilizes all the target system's internal and external SRAM banks and connects all error signals of used SRAMs. More memory improves detection accuracy because more events can be counted, leading to higher statistical confidence and providing a better resolution for the solar condition prediction and optimal mode selection. Moreover, the proposed design can extend the sensing elements if the addition of external devices (e.g. DRAM with EDAC) is available in the target system. However, the designer should be aware that these devices' sensitivity to radiation and characteristics are not the same as SRAMs. For example, DRAMs usually have a lower upset rate compared to SRAMs.

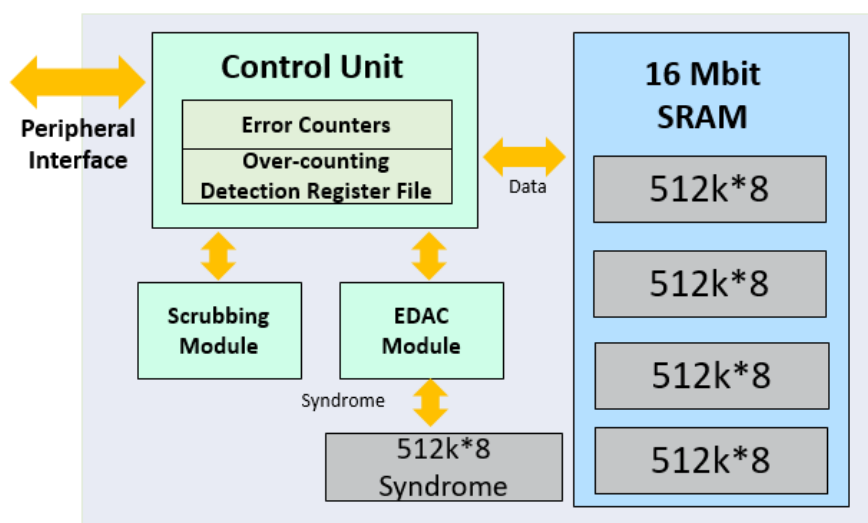
### 4.3 Implementation of the SEU Monitor

As shown in Figure 4.1, the proposed SEU monitor can be embedded into internal or external sensing elements. In general, the realization of these two conditions is straightforward and similar. However, by considering the actual application, several differences and requirements need to be

considered for the implementable target design, such as timing, communication, misaligned access, etc. In this section, the implementation of the proposed monitor on the external Synchronous SRAM and internal SRAM based on one RISC-V SoC architecture has been introduced in detail.

### 4.3.1 External SRAM-based implementation

For the evaluation study, the proposed SEU monitor has been integrated with a scalable and configurable master-slave architecture 20-Mbit Multi-Chip Module (MCM) SRAM [SSC<sup>+</sup>19], as shown in Figure 4.8. The SRAM architecture is asynchronous, which is inherently more durable than clock-based designs since the clock tree is typically a weak point in integrated circuits and bit faults can spread across the device. The synchronous interface, on the other hand, is required for the controller to access the asynchronous memory blocks. The result is a 16 Mbit MCM Synchronous SRAM (SSRAM). This self-contained SSRAM is made up of five identical 512 k \* 8 chips. Four of them serve as slave modules, forming a 16 Mbit data storage space, while one serves as the master, storing the syndrome data. Moreover, the proposed dedicated detection flow is embedded into the main control logic to manipulate the EDAC, scrubbing, ODRF, fault counters, and detection flow mechanisms. The memory blocks are based on the conventional 6T memory cell shown in Figure 4.2. Furthermore, the number of slave modules in the MCM can be increased from 4 to 8, 16, or even more, making this MCM SRAM architecture scalable. As a result, a sufficiently large sensitivity zone can be achieved, guaranteeing a high probability of particle detection.



**Figure 4.8:** Master-slave architecture of the 20-Mbit MCM SSRAM with SEU monitor.

Each 32-bit data block is protected by the HSIAO (39, 32) SEC-DED code. EDAC calculates a 7-bit parity syndrome during the write operation and stores it in the internal 4-Mbit SRAM.

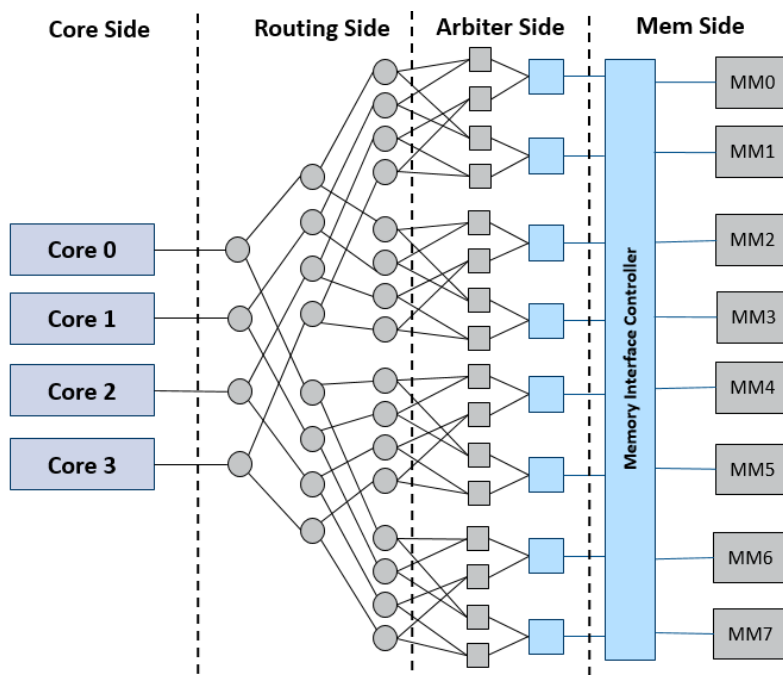
During read operation and SEU detection operation, the data is read from the slaves, which also includes the reading of the corresponding syndrome. Scrubbing, on the other hand, is accomplished by performing periodic read operations, fixing single-bit faults using EDAC, and progressively copying data back into memory for each memory location. Scrubbing is completely autonomous and transparent to the user, which means that the user can access the SRAM even while the operation is running. The scrubbing model can periodically read memory words when the chip is idle, or the fault detection flow is active, and the scrubbing address can be automatically incremented after completing the current scrubbing cycle. The addresses start from zero to the last  $2^{19}-1$  address, after which it starts again from zero. The scrubbing rate, which is the delay between accessing consecutive memory words, can be configured by the user, but the operating frequency limits it. In [SSC<sup>+</sup>19], the test frequency of this MCM SSRAM is 50 MHz, and the minimum scrubbing rate is four clock cycles. Therefore, the minimum time for scrubbing all the memory words is around 42 ms.

A 32 \* 21-bit ODRF and three 8-bit fault counters, as well as the proposed detection flow, are embedded into control logic. A single 21-bit ODRF entry consists of a valid entry bit, a 19-bit address, and an error type bit that differentiates between double-bit and permanent errors. So up to 32 error addresses can be logged simultaneously. If the register file is overflowing, the oldest individual record can be automatically discarded, and a corresponding overflow bit can be set in the status register. Moreover, a valid entry bit can be reset if the user rewrites a double-bit error address. These three fault counters are used to load the number of different errors detected in each period. Moreover, in order to reduce the cost of the SEU monitor, the SEU management unit is not implemented in the external SRAM-based case. Therefore, the results of these fault counters should be managed by the user and transmitted to the host system for further processing. Furthermore, since the sequential logic in the control unit, EDAC, and scrubbing modules can also be intrinsically sensitive to radiation-induced effects, TMR flip-flops can be used to harden the corresponding logic [VPK15].

### 4.3.2 Internal SRAM-based implementation

Unlike implementing an SEU monitor based on an external independent SRAM, it is more complicated to integrate the proposed design into the internal SRAMs of the SoC system. The main reason is that more factors need to be considered, such as communication, timing, unaligned access, etc. The proposed SEU monitor has been implemented into a multiprocessing system, as shown in Figure 6.4. The target multiprocessor is a quad-core SoC based on 32-bit RI5CY cores, which is extended from the open-source single-core microcontroller architecture PULPissimo [PULP] of the ETH Zurich and the University of Bologna. The target SoC represents a significant step ahead in complexity and robustness compared with the baseline PULPissimo platform. The target multiprocessor supports the parallel execution of the quad-core system, core-level self-adaptive fault tolerance (detailed in Section 6.2.1), and in-flight real-time SRAM SEU detection

and prediction. The target RI5CY core [MGA<sup>+</sup>17] is an in-order, single-issue core with four pipeline stages and full support for the RV32I (base integer instruction set), RV32C (compressed instruction set), and RV32M (multiplication instruction set extension) instructions. The RI5CY core is intended to boost the energy efficiency of low-power signal processing applications.

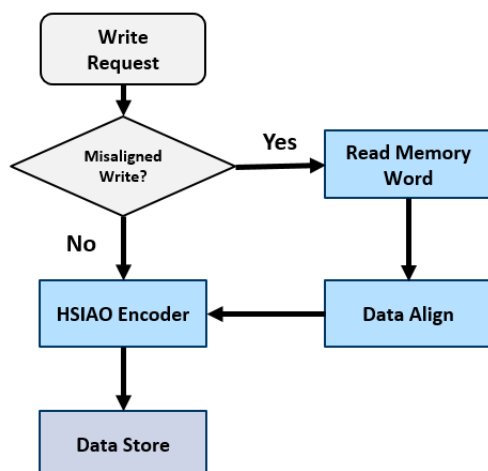


**Figure 4.9:** Mesh of trees 4\*8 communication architecture for a quad-core system with eight internal SRAM banks. The proposed monitor is integrated into the light blue parts.

A total of 8 different 8192 x 40-bit shared L2 SRAM banks are used for building this tightly-coupled multiprocessing architecture. The instructions and data for each core are stored in different memory banks. Each memory word has 40 bits, of which 32 bits are used for data storage, and the remaining 8 bits are used to store the corresponding EDAC parity bit. The Tightly Coupled Memory Interconnect (TCMI) [RLR<sup>+</sup>11] is a fully integrated parametric Mesh-of-Trees (MoT) interconnection network designed to enable high-performance, single-cycle communication between processors and memory banks. To provide equal access to memory banks, the TCMI network employs round-robin arbitration and fine-grained address interleaving. The TCMI of fundamental building blocks, as illustrated in Figure 4.9, contains a combinational path: routing primitives (circles blocks) and arbitration primitives (square blocks). The routing primitive is used to build individual routing trees that link the cores to the arbitration tree (and vice-versa). Arbitration trees are used to handle concurrent requests and route them to memory ports or vice versa. Because of the unique architecture of TCMI, a single-cycle latency system can be created, and the cores may access memory banks in parallel effectively.

The proposed SEU monitor system is integrated into the memory interface controller and the arbitration tree, which could take control of the access of mem banks. Moreover, the proposed SEU monitor (i.e., EDAC, scrubbing, detection flow, etc.) is required for each memory block. The main reason for not using a single SEU monitor is to support core parallel access SRAM banks. Therefore, each memory bank can be protected by its own EDAC module (i.e. HSIAO (39, 32) SEC-DED). During the write operation of cores, the cores first select the target memory banks through the TCMI network. Then, the EDAC module in the bank can calculate the corresponding 7-bit parity. The syndrome can be combined with the input 32-bit data, plus an invalid bit, and stored together into the target 40-bit memory word. The above-mentioned writing process is completed in a combinational logic circuit. In order to ensure the correctness of the timing sequence, these need to be completed within a single cycle. During the read operation, the data is read from the corresponding memory word, then decoded by the EDAC module, and sent back to the target core through the TCMI network.

The SEU detection process can be activated regularly by the SoC, or it can also be triggered by directly sending the activity signal through the cores. Since each module works independently, different modules can be activated for SEU detection. Therefore, the sensing element area can be configurable. After triggering the SEU detection on one SRAM bank, the detection flow, as shown in Figure 4.5, can be active. After that, the scrubbing module starts to access the memory words, and the detected errors can be loaded into the corresponding fault counters. It should be noted that during the operation of the SEU monitor, the corresponding SRAM banks cannot be accessed by the cores. Therefore, the corresponding arbitration primitives in the TCMI network arbitration tree can not respond to the access request from the cores during the scrubbing period.



**Figure 4.10:** Misaligned data access procedure in the memory interface controller.

Since the SEU monitor is added between the core and SRAM, in addition to the timing, attention should also be paid to the impact of the cores on SRAM banks access mode. The Load Store Unit



(LSU) in the target RI5CY core is responsible for accessing the data memory, and the LSU can perform unaligned accesses, meaning memory bank accesses are not aligned on natural word boundaries. Therefore, during each read or write request, the RI5CY cores may load or store words (32 bits), half-words (16 bits), and bytes (8 bits) on the memory banks. Furthermore, the LSU must execute two separate word-aligned accesses internally, requiring at least two cycles for misaligned loads and stores in the RI5CY core. In order to achieve the misaligned access, the core needs to send a byte to enable a control signal to the target memory banks for setting the bytes to write or read. The misaligned access would affect the correct operation of the SEU monitor, mainly because directly using the EDAC module to encode the half-word and byte data could generate the wrong syndrome during the misaligned writing process. One possible solution is to add a corresponding encoding mechanism for each type of unaligned access. However, compared to the case of one encoder, this causes much overhead. The other solution is to convert the misaligned data into the aligned data before sending it to the encoder, as shown in Figure 4.10, and a writing control logic is integrated into each memory bank. After receiving the normal writing request, the incoming 32-bit data can be directly encoded by the HSIAO encoder and loaded into the corresponding memory word in the same clock cycle. On the other hand, after receiving the misaligned write request, the original data of the target memory word can be read out first. Then, according to the byte enable control signal, the incoming and original data are combined into new aligned 32-bit data. Finally, the newly aligned data can be encoded by the HSIAO encoder and then stored back into the corresponding memory word. The aforementioned method can ensure the normal operation of the SEU monitor without adding additional hardware. However, an additional clock cycle is required to achieve the data alignment. Moreover, in order to avoid the timing conflict during the misaligned writing data alignment stage, the arbitration primitives for the corresponding memory block in the TCMI should not respond to any access requests. Therefore, each misaligned writing requires two cycles, and the core must pause for one cycle to perform the following operations. Thus, the performance of the RI5CY core would be a little bit reduced. By testing several commonly used benchmarks, the situation of misaligned writing does not happen very often. For example, this would only happen dozens of times for the well-known Dhrystone benchmark [BDH].

The SEU management unit is embedded into the framework controller, as shown in Figure 6.4 and detailed in Section 6.2, which intends to collect all the fault counter data. Since the multiprocessing systems have an inherent hardware redundancy and are convenient for deploying reconfigurable mechanisms, such as the core-level N-Module Redundancy (NMR), dynamic voltage and frequency scaling, etc. The data in the SEU management unit can be used to trigger the self-adaptive properties of the SoC. Therefore, enabling the adaptive trade-off between reliability, power consumption and performance, the details described in Chapter 6. Moreover, the data in the SEU management can also be used to predict high-energy particle explosion events, i.e. SPEs, which the details are provided in Chapter 5.

## 4.4 Analysis of Results

### 4.4.1 SEU Monitor Performance Analysis

The main target of the proposed monitor is to detect and count the radiation-induced faults in the target sensing element, i.e. SRAM banks. A set of simulations is performed on the designs described above by considering different numbers of faults in each memory word and evaluating the proposed monitor's effectiveness. The test procedure includes randomly injecting many soft, or hard bit flips in the target memory banks, corresponding to SEUs, MCUs or permanent faults. The result shows that for the utilized HSIAO (39, 32) code-based fault detection, all single and double-bit soft faults as well as the single-bit hard faults in each memory word can be correctly detected and counted during the checking process. However, due to the limitation of the selected HSIAO SEC-DED code, MCUs in the same memory word may not be correctly detected.

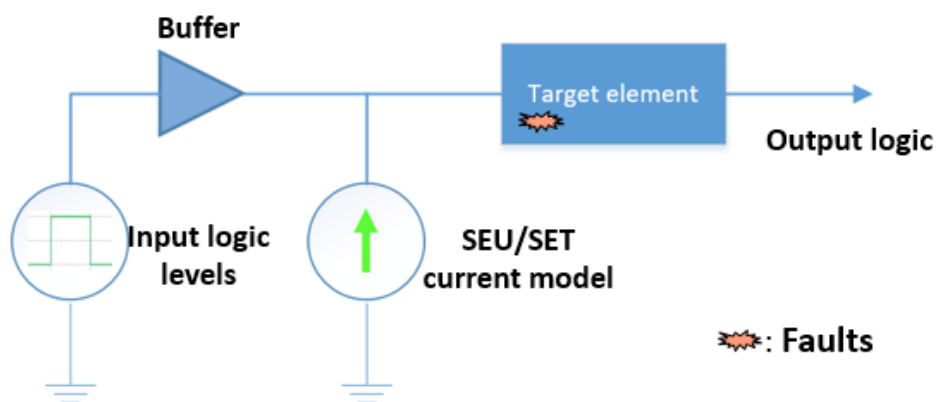
The MCUs are an essential phenomenon and are also considered in our previous SEU monitor design. The HSIAO SEC-DED cannot correctly handle multiple bit errors (bit errors  $\geq 3$ ) in each word. The odd number of bit flips can be detected as single-bit errors, and the even number of bit flips can be seen as double-bit errors. Generally, the traditional codes may not provide good detection and correction properties for MCUs. The MCUs can be corrected or detected by implementing suitable EDAC codes, such as SEC-DED (Single Error Correction-Double Error Detection) [MYH70], SEC-DAEC-TAEC (Double Adjacent Error Correction-Triple Adjacent Error Correction) [SPG<sup>+</sup>15], 3-bit burst ECC [LRX<sup>+</sup>19]. These codes are very convenient for the correction of multiple adjacent errors (up to three adjacent bit-flips). However, these codes usually incur a higher overhead of area and timing compared to the HSIAO SEC-DED code, and the corresponding hardware implementation of EDAC is also more complicated. For example, as shown in [LRX<sup>+</sup>19], compared to the SEC-DED, the 3-bit correction code incurs 2x area and 3x power overhead for word sizes of 128 bits or larger. Therefore, the proposed monitor aims to perform the trade-off between achieved robustness and the overheads, and based on this, the optimal solution for the mitigation of MCUs could be chosen. Thus, it is possible to overcome the MCU issue by applying more complex EDAC codes with a similar detection flow. Moreover, the well-known interleaving technique can also be applied to provide further protection against multiple-bit errors in each memory word. This technique can distribute the memory cells from the same word into different columns so that they are physically distant from each other, and the probability that a single particle hits multiple bits of the same word is drastically reduced. Thus, MCUs in different memory words can be corrected or detected by the EDAC code, and the MCUs in the same memory word can also be mitigated by using the interleaved SRAM technology. In general, the selection of the HSIAO code provides fast and straightforward encoding/decoding features with low hardware overhead. Moreover, as the probability of adjacent double-bit errors is much higher than other MCUs in the memory word, thus, the HSIAO SEC-DED code is a suitable choice for the SEU monitor design. In addition, the occurrence probability of uncorrectable MCUs

and the accumulation of transient faults can also be significantly decreased if the entire memory is scrubbed regularly [SSM00].

Regarding permanent faults, the proposed detection flow can log the corresponding fault address into the ODRF. Thus, the user can easily access this info and do the necessary masking. Moreover, it is possible to use the other approach to detect permanent faults of the sensing elements relying on the implemented mechanisms. One of these methods is to report single-bit errors by raising the error output pin. The control unit has a field in the status register which tells the erroneous bit position. By writing and reading data patterns at that address, the software can determine the presence of permanent faults. However, the more sophisticated approach is described in Subsection 4.2.3, which was also verified by a set of simulations.

#### 4.4.2 SEU Sensitivity Analysis

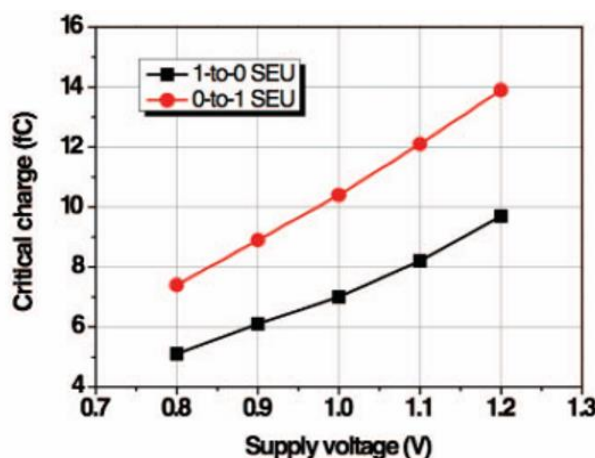
The SEU monitor should be more sensitive to particle strikes than the standard logic gates, which constitute the target design. Since the SRAM is intended to operate as a radiation monitor within a host chip designed with standard logic gates, it is essential to evaluate the SEU sensitivity of the SRAM cell with respect to the SEU sensitivity of standard flip-flops and the SET sensitivity of standard combination gates. Moreover, it is also essential to evaluate the SEU sensitivity of the SRAM in dependence on supply voltage since low-power (e.g. voltage-scaling, detailed in Section 6.2.2) techniques could be applied in space applications.



**Figure 4.11: Block diagram of the standard current injection approach in SPICE simulation to evaluate target element SET/SEU critical charge.**

SET/SEU sensitivity has been evaluated in terms of critical charge which was estimated through the standard current injection approach in SPICE simulations (as shown in Figure 4.11), i.e., by injecting the double-exponential current pulse [FDT<sup>+</sup>13] in the circuit nodes. The constant timing parameters of the double-exponential current pulse were used (rise time = 10 ps and fall time = 100 ps), while the injected charge was varied during the simulations to obtain the critical charge values. Figure 4.12 depicts the variation of the critical charge of an SRAM cell in terms of supply

voltage for the cases when a logic '0' and a logic '1' are stored in the cell. It is essential to mention that the analyzed SRAM in this section is selected from IHP 130 nm technology. Although the current injection experiments in this section use different technology than the target SRAM, as introduced in Section 4.1, the results are still significant for analyzing the SEU sensitivity variety since the proposed monitor is general and can be implemented in different technologies, with various sensing elements. It can be observed that the critical charge depends on the stored value, and it decreases as the supply voltage is reduced. The reduction of supply voltage leads to a decrease of the driving strength of transistors, consequently reducing the transistors' capability to dissipate the induced charge.



**Figure 4.12: Example of the SRAM cell's critical charge in supply voltage during current injection experiment in SPICE simulation.**

The critical charge values for the analyzed SRAM and the most common standard gates in the IHP 130nm CMOS library, obtained for the nominal supply voltage of 1.2 V, are presented in Table II. Since the critical charge for combinational gates depends on the input logic levels, only the lowest critical charge values for each gate, obtained by injecting the current pulse at the output of the gate, are presented. As can be seen, for the logic '1' stored in the SRAM cell, the critical charge of the SRAM cell is lower than that of all investigated standard cells. On the other hand, when the logic '0' is stored, the critical charge of the SRAM cell is slightly higher than that of NOR, XOR and XNOR gates. However, since a charge higher than the critical charge is required to cause a SET capable of propagating through the combinational circuit, it is clear that the SRAM cell is more sensitive to particle strikes than all investigated logic cells. This indicates that the analyzed SRAM cell can be utilized as a radiation monitor within a system designed in the investigated 130nm CMOS technology. Moreover, this analysis relates to the regular standard cell library. In the radiation-hardened library, both standard cells and SRAMs are additionally hardened. Nevertheless, due to the cost of hardening, the critical charge of SRAM cannot be effectively increased very much.

**Table 4.7:** Critical charge for SRAM and different standard cells.

Element	Critical Charge (fC) for SET or SEU
SRAM (stored logic "1")	9.7
SRAM (stored logic "0")	13.9
D flip-flop	24.9
INV	20.1
NAND	22.2
AND	19.6
NOR	12.8
OR	19.3
XOR	12.9
XNOR	13.6

#### 4.4.3 Synthesis Results

Since the SEU monitor is implemented with the internal or external stand-alone SRAM banks, it is essential to investigate the introduced power/area overhead. As an example, the following synthesis results are based on the IHP 130 nm standard CMOS library with a 20-Mbit SRAM under a supply voltage of 1.2 V and a nominal operating frequency of 50 MHz. After performing the synthesis analysis, the total area of the 20 Mbit SRAM with the proposed monitor is around 14 mm<sup>2</sup>, and the corresponding power consumption is 384 mW. Regarding area occupation, the main contributor is the 20 Mbit SRAM blocks, while the contribution of the entire "digital" logic (control unit with the detection flow, ODRF, EDAC, Scrubbing module) is only 0.0957mm<sup>2</sup>, i.e., the introduced area overhead is less than 1%. Similarly, the estimated power consumption is only 0.211 mW, i.e., the induced power overhead is even less than 0.1%. Moreover, the area and the power cost of the proposed monitor can keep the same when integrating with larger SRAM banks. Thus, the induced corresponding area and power consumption ratio can be much lower when applying the proposed design with considerable sensing elements. Therefore, the area and power overheads resulting from additional modules are negligible.

**Table 4.8:** Area comparison (in  $\mu\text{m}^2$ ) between radiation-hardened controllers with and without proposed SEU monitor.

	<b>Without Monitor</b>	<b>With monitor</b>
<b>Combinational area</b>	11298	67344
<b>Non-combinational area</b>	7408	28395
<b>Total area</b>	18706	95739

**Table 4.9:** Power consumption comparison (in mW) between radiation-hardened controllers with and without proposed SEU monitor.

	<b>Without Monitor</b>	<b>With monitor</b>
<b>Combinational area</b>	0.022	0.091
<b>Non-combinational area</b>	0.032	0.120
<b>Total area</b>	0.054	0.211

Moreover, it is also important to compare only the "digital" parts of the above example chip with and without the SEU Monitor. Tables III and IV show such a comparison. The main parts of the SEU Monitor are the three 8-bit rad-hard counters and the  $32 \times 21$ -bit address register file. The number of flip-flops in the proposed design is 947, while the original design without SEU Monitor has 253 flip-flops. Therefore, these additional flip-flops are the main contributors to the power and area overhead.

## 4.5 Summary

In this chapter, the on-chip low-cost SEU monitor for space applications, based on the internal or external SRAMs as sensing elements, has been presented. The idea is to employ the standard on-chip SRAM memory as a particle detector parallels to its normal data storage function. The SRAMs on the embedded system can detect faults caused by high-energy particles, and the sensitivity of 65 nm SRAM banks for radiation-induced faults has been analyzed in detail. The proposed design extends the basic functionality of the SRAM banks by integrating the EDAC, scrubbing, a dedicated detection flow, and faults management units. The dedicated detection flow is the crucial part of the proposed monitor, which controls other modules to perform memory word in-order checks, prevent fault over-counting, and perform fault classification. Therefore, the SEUs and MCUs, as well as the permanent faults, in each memory word can be accurately detected and counted. The proposed monitor can provide accurate in-flight radiation-induced fault measurements for the target SRAM during run-time.

Two example implementations of the proposed monitor with external stand-alone SRAM banks and internal memory banks in a quad-core system have also been introduced. With a proper design, the integration of the proposed monitor can be easily achieved without affecting the other existing functions, such as data read/write, misaligned data access, etc. The sensitivity comparison of the SRAM cells with the other standard logic gates has been performed with the standard current injection approach. The SPICE simulations confirmed that the SRAM cells are susceptible to particle strikes and suitable for detecting faults. The synthesis results show that the induced area and power consumption overheads of the proposed monitor for the host system are negligible.

The collected in-flight fault data from the proposed monitor is intended to be used to predict the space environment and then determine the optimal configuration mode of fault-resilient systems. Therefore, the accurate counting of radiation-induced in-flight faults on the target sensing elements is a vital cornerstone work. The proposed monitor is critical to realizing a self-adaptive resilient system for achieving the best trade-off between reliability, performance, and power consumption during run-time.

## Chapter 5

# Solar Particle Event Prediction

The SPEs can dominate the space radiation environment and increase the probability of radiation-induced faults, especially SEUs, in space-borne electronic systems by several orders of magnitude. Therefore, it is vital to enable the early detection of SEU rate changes to ensure the timely activation of dynamic radiation hardening measures. This chapter introduces an embedded approach for predicting SPEs by combining real-time SEU measurement, an offline-trained machine learning model, online learning prediction optimization, and a dedicated hardware accelerator. With respect to the state-of-the-art, the proposed method brings the following benefits: (1) Prediction of radiation environment one hour in advance, with fine-grained hourly tracking of SRAM SEU rate variations during SPEs as well as under normal conditions; (2) Online optimization of the prediction model for enhancing prediction accuracy during run-time; (3) Negligible cost of hardware accelerator design for the implementation of selected machine learning model and online learning algorithm. The proposed design is intended for a highly dependable and self-aware resilient system employed in space applications, allowing to trigger the appropriate radiation mitigation mechanisms before the onset of high radiation levels.

The chapter is divided into six sections. Section 5.1 presents the concept of the prediction method. The historical solar events analyzing procedure is detailed in Section 5.2. The training and evaluation for supervised machine learning and online learning algorithms are presented in Section 5.3. Section 5.4 introduces the architecture of the dedicated hardware accelerator. The design analysis is presented in Section 5.5. The summary is presented in Section 5.6.

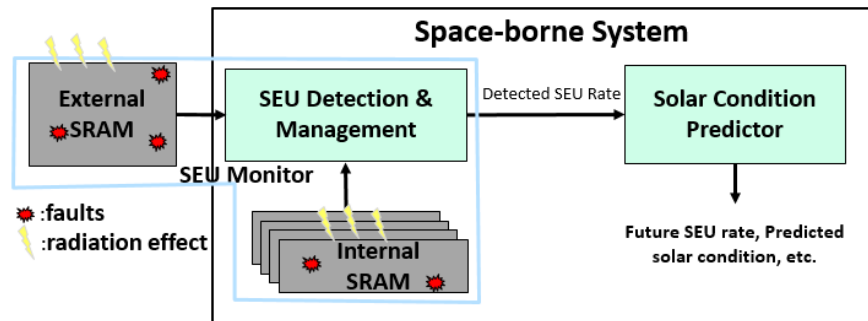
### 5.1 Concept

As introduced in Section 2.2.1, one of the main causes of SEUs in space is the Solar Particle Event (SPE) phenomenon. During an SPE, a large number of energetic particles are emitted into space, and this event can last from several hours up to several days. These energetic particles can induce SEUs either by direct ionization or indirect ionization [DWM03]. Since the particle flux directly



determines the SEU rate of an electronic system, the SEU rate may increase hundreds to thousands of times during the SPE peak periods. Thus, it is essential to track the variation in particle flux in real-time and consequently activate the suitable mitigation techniques to protect the sensitive elements in onboard electronic systems.

Furthermore, in order to perform efficiently SEU mitigation and sustain system operation, real-time flux fluctuation prediction, i.e. forecasting forthcoming SPEs, is required. In such a way, radiation mitigation techniques can be deployed before the outburst of particles during SPEs. In order to facilitate the SPE prediction based on real-time SEU detection, various machine learning algorithms can be applied to predict SEU variations. However, according to the introduction in Section 3.2, there isn't any publically accessible work on the application of machine learning algorithms to forecast SPEs and SEUs from in-flight observed SEU data, which can help self-adaptive mechanisms in space-borne systems.

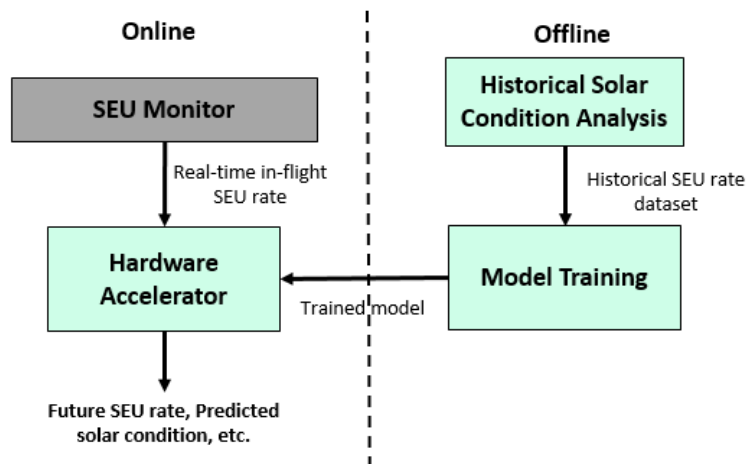


**Figure 5.1:** General concept of the SEU monitor with solar condition predictor inside a space-borne system. The green blocks represent the main focus of this dissertation.

Figure 5.1 presents the concept of an SEU monitor network with the solar condition prediction unit in a space-borne system. As introduced in Chapter 4, the SEU monitor provides the pre-processed in-flight real-time SEU information, which the solar condition predictor can use for analyzing the future radiation environment. The solar condition predictor applies collected in-flight real-time SEU rate data and a pre-defined prediction model to forecast the following SEU rate, thus, determining the corresponding solar conditions. This thesis describes the corresponding solar condition predictor as a supervised machine learning problem. Moreover, the outputs of the proposed predictor can be used to determine the state of a self-adaptive processing system. Based on this, the optimal selection of mitigation solutions under variable radiation conditions can be achieved. The significance of the design is that the capacity to foresee the increase in radiation levels reduces the possibility that the target system can be exposed to harsh environments without adequate protection.

The purpose of the proposed solar condition model is to utilize the introduced embedded on-chip SRAM-based SEU monitor, the supervised machine learning model, and historical solar

events flux data to forecast the in-flight SRAM SEUs and the occurrence of the SPE. This solution supports a pre-trained regression model to provide a fine-grained prediction of the SEUs at least 1 hour in advance. Moreover, an online learning method is also implemented, which can further improve the prediction accuracy of machine learning in real-time. A low-cost hardware accelerator is customized to execute the proposed machine learning prediction model and online learning algorithm. The proposed hardware accelerator's cost and area/power overheads are negligible compared to the host SRAM. The main reason for using the prediction with the SEU measurement, rather than just measuring the SEU rate, is to minimize the possible adverse impact of radiation on the system operation. Namely, if only the SEU measurement is employed, it may be too late to react once the SEU is detected because the monitor needs a specific time period to collect and process the information.



**Figure 5.2:** General flow of the proposed solar condition prediction method. The green blocks represent the main focus of this chapter.

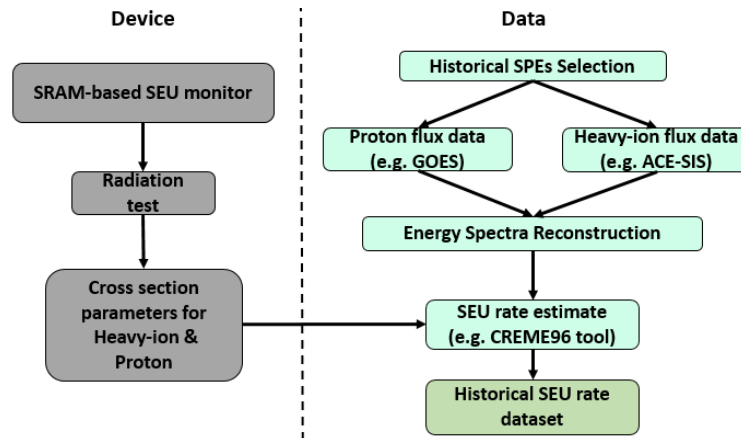
In Figure 5.2, the evaluation flow chart of the solar condition prediction flow is illustrated. The proposed flow consists of two phases:

*Offline phase* - applying historical space flux data (from previous space missions) to establish a suitable SEU prediction machine learning model.

*Online phase* - measurement of the real-time in-flight SRAM SEU count and predicting upcoming SEU changes.

There are two main blocks in the offline phase: historical solar condition analysis and supervised machine learning model training. In the solar condition analysis block, the in-flight hourly Soft Error Rate (SEU rate) of the target SRAM during several historical solar events is determined, as discussed in Section 5.2. The collected hourly SEU rate is processed for training in the supervised machine learning block. A suitable SEU prediction regression model can be obtained based on offline training, as described in Section 5.3.

The online phase contains two main blocks: real-time SEU detection and hardware accelerator. The real-time SEU detection is performed continuously during the mission, as introduced in Chapter 4. It is essential to mention that the existing SRAM resources are utilized for SEU measurement, which minimizes the area and power overhead. The number of detected in-flight real-time SRAM SEU rates are stored and processed by the hardware accelerator. The hardware accelerator implements the offline trained machine learning model and the online learning algorithm, detailed in Section 5.4. The online learning algorithm is used to enhance the accuracy of the prediction, since the offline training model may not consider all realistic scenarios, as discussed in Section 5.3.4. Therefore, the predicted SEU data can be collected, and the prediction accuracy improvement can also be achieved during the run-time. Moreover, the functionality of online phase components may be affected by high-energy particles. Thus, the SEUs, Single Event Functional Interrupts (SEFIs), Single Event Latchup (SEL) or micro-SEL may occur in online phase components. The corresponding appropriate design measures, such as Triple-Module Redundant (TMR) flip-flops [VPK15], are applied to mitigate these effects and enhance the robustness against radiation particles.



**Figure 5.3:** Evaluation process of the historical solar condition analysis model for collecting the SEU rate dataset of the target SRAM-based SEU monitor from past solar events. The green blocks represent the main focus of this chapter.

## 5.2 Historical Solar Condition Analysis

The complete process of obtaining in-flight SRAM SEU rates using actual historical space ion flux data is presented in this section. It is one of the main blocks of the offline phase in the evaluation flow, as shown in Figure 5.2. The entire evaluation flow of the historical solar condition analysis model is depicted in Figure 5.3. The hourly SEU rate of a target SRAM-based SEU monitor is computed in the historical solar condition analysis utilizing an experimentally acquired cross-section of SRAM and the hourly flux database gathered from past space missions. The obtained

SEU rate dataset can be processed and used as training data for the machine learning block. The general historical solar condition analysis process includes the following three steps:

- 1) Historical space flux data collection
- 2) Energy spectra reconstruction
- 3) SEU rate estimation

### 5.2.1 Historical Flux Data Collection

The historical space flux data from previous orbiting satellites is publicly available and can be used to analyze previous solar activities. The solar cycle, as defined in Section 2.1, is a nearly periodic 11-year fluctuation in the Sun's activity. In order to conduct a comprehensive analysis, in this thesis, all events during one solar cycle, i.e. solar cycle 24, are selected and analyzed in detail. Solar Cycle 24 (2008-2019) is the most recently completed solar cycle, which is the 24th since thorough tracking of solar sunspot activity began in 1755. According to the large events statistics from National Oceanic and Atmospheric Administration (NOAA) [NOAA], 36 SPEs which occurred and affected the Earth's environment in the target event are chosen for analysis.

Table 5.1 presents the details of the selected solar events. According to the definition from the NOAA, the start of one SPE is at least three consecutive five-minute intervals of flux data points  $\geq 10 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$  of the proton with energy  $\geq 10 \text{ MeV}$ . In addition, the end of the solar event is defined as the last time the flux  $\geq 10 \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}$ . The flux of soft X-rays measured by the GOES satellite is used to categorize SPE sizes. According to the peak flux ( $\text{W}/\text{m}^2$ ) of 100 to 800 picometer X-rays, many detected SPEs are classed as A, B, C, M, or X [HUS17]. Each class has a tenfold increase in peak flux, with X class flares having a peak flux of order  $10^{-4} \text{ W}/\text{m}^2$ . There is a linear scale from 1 to 9 within a class, thus an X2 flare is twice as potent as an X1 flare and four times as powerful as an M5 flare. The more extreme M and X class flares are frequently connected with a variety of near-Earth space environment impacts, such as disturbance of the Earth's magnetosphere and vivid aurora in the polar regions.

Heavy-ions (direct ionization) and protons (mostly indirect ionization) are the primary causes of radiation-induced effects in space-based electronic systems. Several devices on previous space missions continually monitored the proton and heavy-ion flow in space during selected events. For this thesis, the ions caused by solar events are the primary target. Therefore, the data collected close to the Earth in the heliosphere but outside the Earth's geomagnetic influence is preferred. Thus, the data can avoid the impact of geomagnetically trapped ions and shield protection from the Earth's magnetic field. Moreover, the notable impacts of the South Atlantic Anomaly (SAA), which is an area that leads to an increased flux of energetic particles in this region, can also be neglected in this thesis. In addition, a further requirement was that the space ion data should be easily accessible via the Internet.

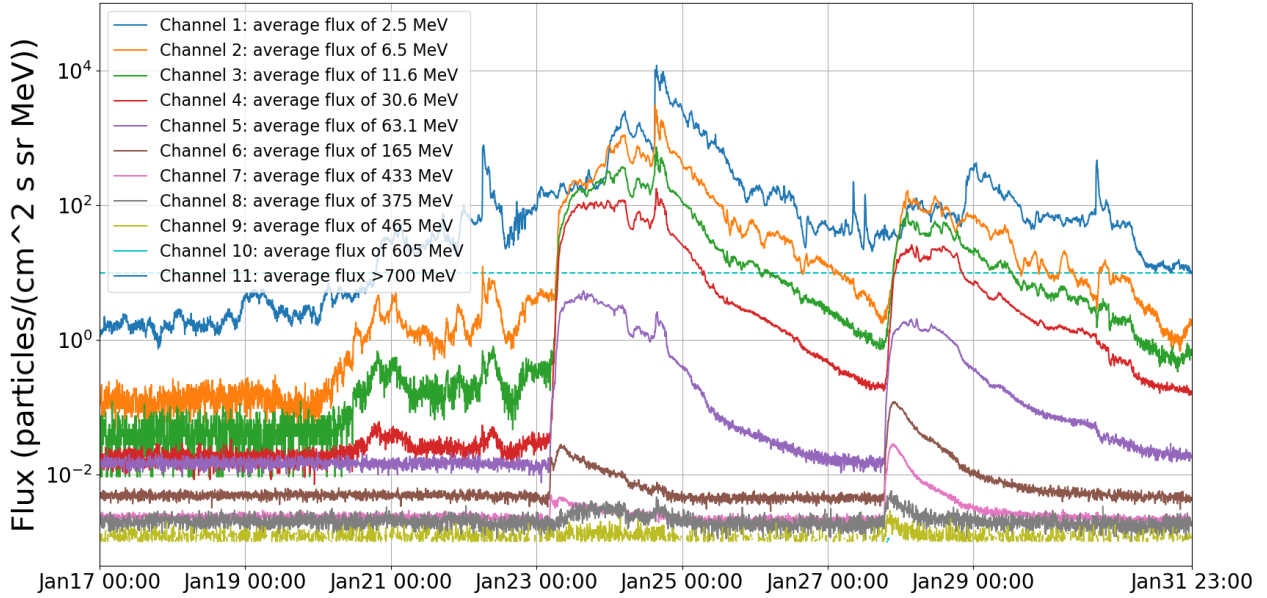
**Table 5.1:** Selected solar event lists from NOAA during solar cycle 24. Proton fluxes are integral 5-minute averages for energies  $> 10$  MeV, given in Particle Flux Units (pfu), measured by GOES spacecraft at Geosynchronous orbit: 1 pfu = 1 p/sq. cm-s-sr. The measured X-rays are used to classify the importance of SPEs.

Event ID	Start Time (Y-M-D UT)	Peak Time (Y-M-D UT)	Proton Flux (pfu@>10MeV)	Importance (Xray)
1	2010-08-14 12:30	2010-08-14 12:45	14	C4
2	2011-03-08 01:05	2011-03-08 08:00	50	M3
3	2011-03-21 19:50	2011-03-22 01:35	14	N/A
4	2011-06-07 08:20	2011-06-07 18:20	72	M2
5	2011-08-04 06:35	2011-08-05 21:50	96	M9
6	2011-08-09 08:45	2011-08-09 12:10	26	X6
7	2011-09-23 22:55	2011-09-26 11:55	35	X1
8	2011-11-26 11:25	2011-11-27 01:25	80	N/A
9	2012-01-23 05:30	2012-01-24 15:30	6310	M8
10	2012-01-27 19:05	2012-01-28 02:50	796	X1
11	2012-03-07 05:10	2012-03-08 11:15	6530	X5
12	2012-03-13 18:10	2012-03-13 20:45	469	M7
13	2012-05-17 02:10	2012-05-17 04:30	255	M5
14	2012-05-27 05:35	2012-05-27 10:45	14	N/A
15	2012-06-16 19:55	2012-06-16 20:20	14	M1
16	2012-07-07 04:00	2012-07-07 07:45	25	X1
17	2012-07-12 18:35	2012-07-12 22:25	96	X1
18	2012-07-17 17:15	2012-07-18 06:00	136	M1
19	2012-07-23 15:45	2012-07-23 21:45	12	N/A
20	2012-09-01 13:35	2012-09-02 08:59	59	C8
21	2012-09-28 03:00	2012-09-28 04:45	28	C3
22	2013-03-16 19:40	2013-03-17 07:00	16	M1
23	2013-04-11 10:55	2013-04-11 16:45	114	M6
24	2013-05-14 13:25	2013-05-17 17:20	41	X1
25	2013-05-22 14:20	2013-05-23 06:50	1660	M5
26	2013-06-23 20:14	2013-06-24 05:20	14	M2
27	2013-09-30 05:05	2013-09-30 20:05	182	N/A

<b>28</b>	2013-12-28 21:50	2013-12-28 23:15	29	C9
<b>29</b>	2014-01-06 09:15	2014-01-06 16:00	42	N/A
<b>30</b>	2014-01-06 09:15	2014-01-09 03:40	1033	X1
<b>31</b>	2014-02-20 08:50	2014-02-20 09:25	22	M3
<b>32</b>	2014-02-25 13:55	2014-02-28 08:45	103	X4
<b>33</b>	2015-06-18 11:35	2015-06-18 14:45	16	M1
<b>34</b>	2015-10-29 05:50	2015-10-29 10:00	23	N/A
<b>35</b>	2016-01-02 04:30	2016-01-02 04:50	21	M2
<b>36</b>	2017-09-05 07:51	2017-09-05 19:30	210	M5

As a result, and due to the easily accessible (publicly available databases), the Geostationary Operational Environmental Satellite-Space Environment Monitor (GOES-SEM) database [GOES], which has continuously provided the data since 1974, has been selected as the proton data source. The Advanced Composition Explorer-Solar Isotope Spectrometer (ACE-SIS) database [ACE], which continuously provides the [He, C, N, O, Ne, Na, Ma, Al, Si, S, Ar, Ca, Fe and Ni] flux data since 1997, has been chosen for the heavy-ion data source.

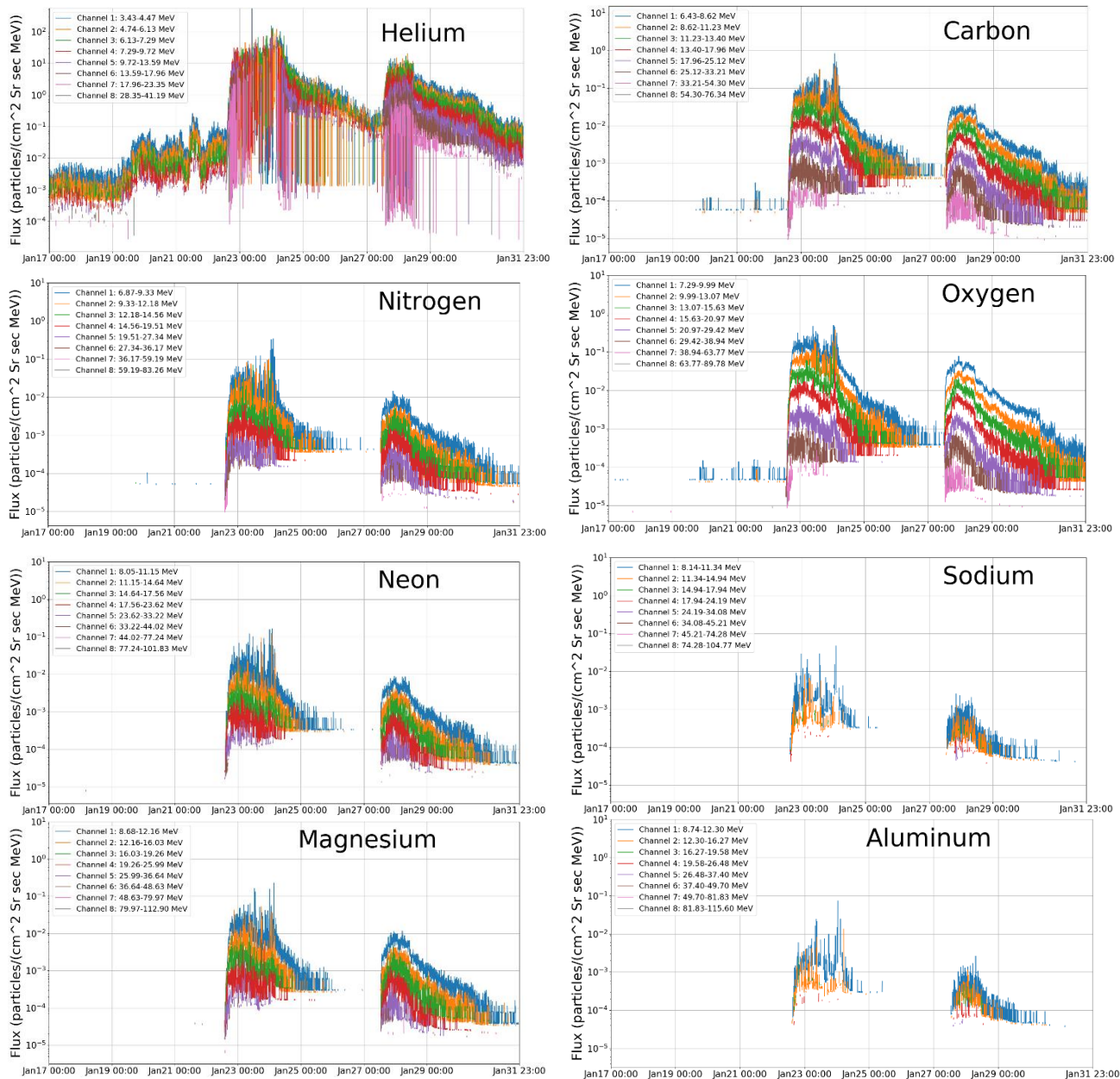
In this thesis, the proton data from the GOES-SEM source [GOES] is used to compute the proton-induced SEU rate. The GOES-SEM database provides continuous processed data on numerous space weather parameters, and it has good coverage of solar occurrences examined in recent decades. The GOES satellites include various subsystems, including three sensors for particle flux observations: two Energetic Proton, Electron, and Alpha Detectors (EPEADs) looking east and west, respectively, and one High Energy Proton and Alpha Detector (HEPAD) pointed to the zenith. EPEADs measure protons with energies ranging from 0.74 to 900 MeV. The HEPAD registers protons with energies  $>330$  MeV. These proton data are from several GOES satellites, which work in the geosynchronous equatorial orbit, and the GOES-13 (launched on 24 May 2006) and GOES-15 (launched on 4 March 2010) is chosen as the proton data sources for this thesis. These two satellites have 11 proton channels, and each detection channel has the same detected average flux. Figure 5.3 illustrates the proton flux from 17 January to 31 2012, where one large and one medium-sized solar event occurred. Moreover, the details of the corresponding 11 proton channels are also shown in Figure 5.4. According to Table 5.1, the start time of these events is "2012-01-23 05:30", in which at least three consecutive five-minute interval flux data points are protons with energy  $\geq 10$  MeV (above the dashed event line)).



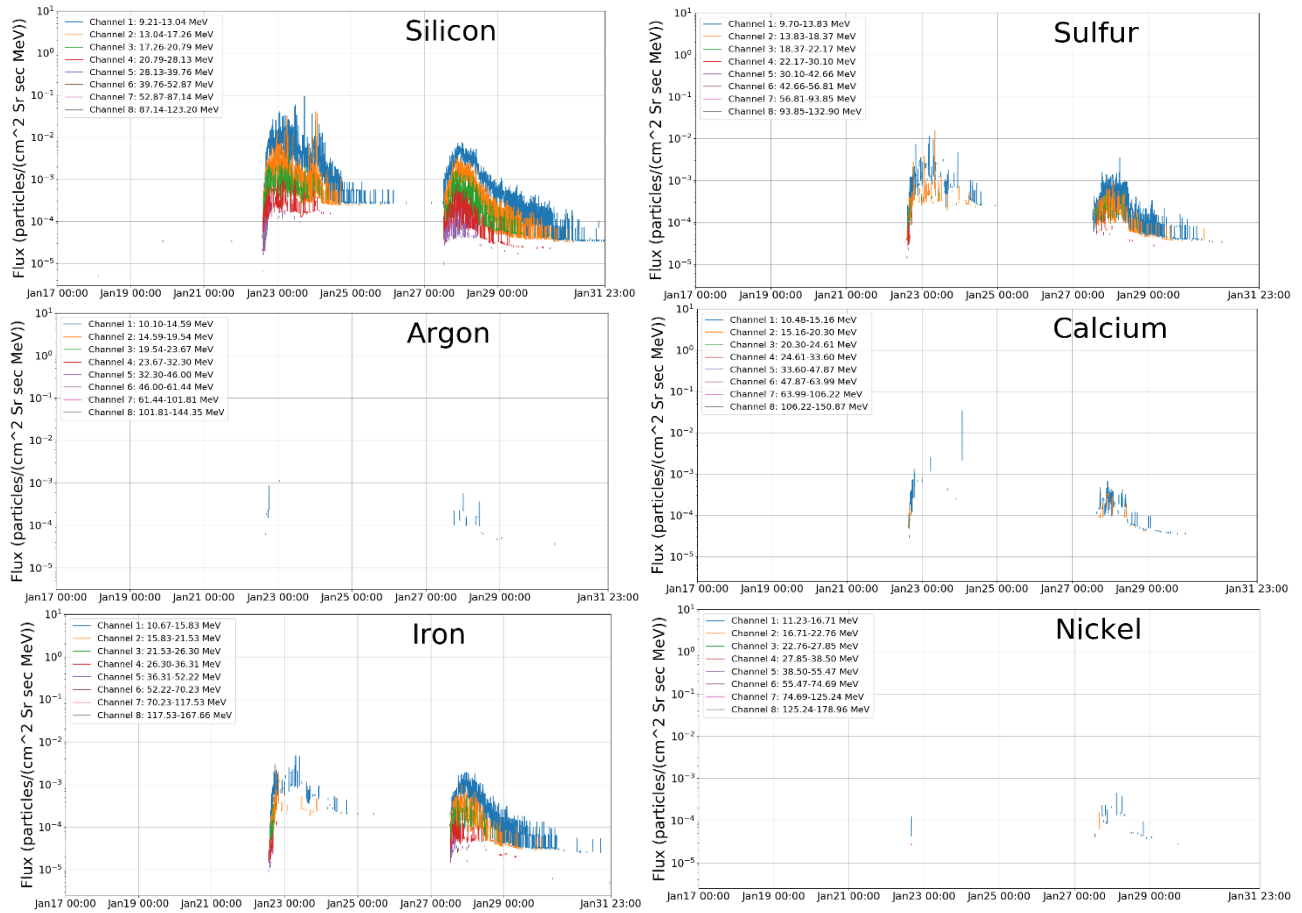
**Figure 5.4:** Example of proton flux during January 17-31, 2012, from GOES-13. Channels 1~7 from the EPEAD detector and 8~11 from the HEPAD detector. The dashed line (the proton with energy 10 MeV) respects the event line.

For SEU rates computed from direct ionization, space heavy-ion flux data from the ACE-SIS source [ACE] is used. On August 25, 1997, the ACE robotic spacecraft was launched and operated in a Lissajous orbit near the L1 Lagrangian point. The SIS detector measures the isotopic composition of energetic nuclei with great precision for various ions with energies ranging from 5 to 150 MeV/nucleon. SIS studies the isotopic composition of the solar corona during large solar flares, when particle fluxes can rise by factors of up to 10000 above quiet-time values. SIS, on the other hand, examines the isotopes of low-energy galactic cosmic rays as well as the composition of anomalous cosmic rays, which are considered to originate in the surrounding interstellar medium, during quiet solar periods. Furthermore, SIS level 2 data arranged into 27-day time intervals is utilised in this research (Bartels Rotations - roughly one solar rotation period). The level 2 data for each Bartels Rotation provides temporal averages of energetically charged particle fluxes for the following time periods: - 256 seconds - hourly - daily - 27 days (1 Bartels rotation). Despite the fact that the ACE-SIS data have been chosen as the reference dataset for computing heavy-ion-induced SEUs, the energy range for ions is rather low, there are significant data gaps, and the data quality of high-energy channels could be poor. Moreover, due to saturation problems in the ion detectors, the space weather monitors of SIS may not function well during solar events [HED04]. Figure 5.5 depicts all ion data with associated channel energy ranges obtained by the ACE-SIS detectors from 17 January to 31 January 2012. Several ions have data gaps and poor data quality, which must be addressed during the SPE energy spectra reconstruction step, as explained in Section 5.2.2. Furthermore, the ACE database contains protons, but the energy range is limited, making it unsuitable for this study.

Furthermore, to ensure the completeness of the gathered space flux dataset, a comparable quantity of proton and heavy-ion flux data acquired during non-SPE times is also collected as a reference. A total of 5107 hourly proton and heavy-ion flow data are gathered in this thesis.







**Figure 5.5:** Example of heavy-ion flux during January 17-31, 2012 from ACE-SIS. The detection channel of each ion is also present in each corresponding sub-figures. Some data gaps are encountered for a few ions.

## 5.2.2 SPE Energy Spectra Reconstruction

The space ion flux data obtained from online databases, as introduced in Section 5.2.1, cannot be directly used for the SEU rate estimation due to: (i) limited types of detected heavy ions, (ii) insufficient energy range, and (c) ion flux data gaps. The common approach to solving the problems above described is to generate the mission flux data from composition ratios with existing flux information [HED04]. The differential flux for each ion needs to be processed to obtain the energy spectrum, which is required for the SEU rate estimation. This thesis aims to reconstruct the hourly average energy spectrum for the ions and protons, which is a total of 5107 hours for the selected events.

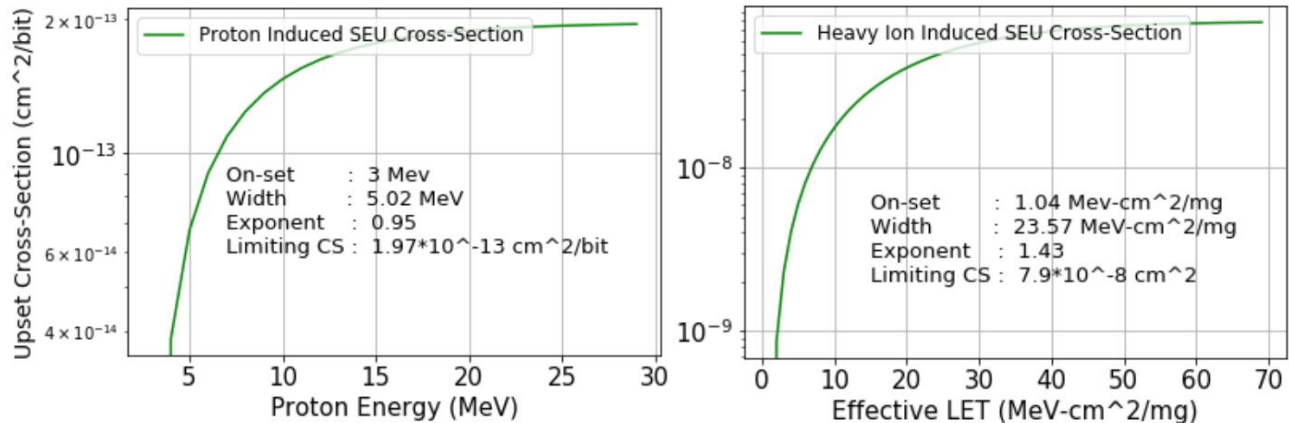
In this thesis, a widely used suite, Cosmic Ray Effects on Micro-Electronics 96 (CREME 96) [AJT97] [CRE96], is utilized to create numerical models of the ionizing-radiation environment in

near-Earth orbits and aid in analysis and verification. CREME96 is one of several tools available in the aerospace industry to provide accurate space environment calculations. Over the years since its introduction, the CREME 96 model has been compared with on-orbit data and demonstrated its accuracy. Moreover, the SPE models in the CREME 96 are based on the actual measurements of the October 1989 solar event, which is undoubtedly one of the most significant particle events since August 1972. The CREME96 96 outfit includes three solar particle models (Peak Five Minutes (P5M), Worst Week (WW), and Worst Day (WD), all of which are based on GOES proton data and high-energy heavy-ion observations from Chicago's IMP-8/CRT. Furthermore, the proton and heavy-ion measurements for the CREME96 96 SPE model event ranged from 500 to 800 MeV, depending on the species.

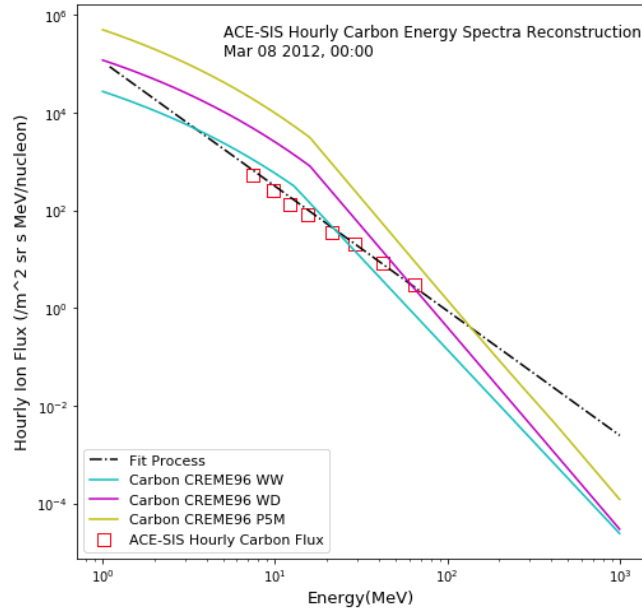
**Table 5.2:** Target SRAM SEU rate (upsets/bit/day) sensitivity to ions with different existing CREME 96 SPE models.

Ions	Worst Weeks	Worst Day	Worst Five Minutes
He(2)-U(92)	$6.97357 \times 10^{-5}$	$3.20759 \times 10^{-4}$	$1.19209 \times 10^{-3}$
ACE-SIS	$6.97202 \times 10^{-5}$	$3.20716 \times 10^{-4}$	$1.19194 \times 10^{-3}$

In order to perform the estimation of the target SRAM-based SEU monitor with the CREME 96 tool suite, the corresponding proton and heavy-ion induced cross-section data are necessary. As introduced in Section 4.2.2, the target SRAM in this thesis is the COTS SRAM from Cypress, designed in 65 nm bulk CMOS technology. Figure 5.6 presents the Weibull fit for heavy-ion and proton static bit cross-section curves for the target SRAM obtained from corresponding radiation tests [VGU17].



**Figure 5.6:** Weibull fit of the target 65 nm SRAM heavy-ion and proton static bit cross-section results from radiation tests [VGU17].



**Figure 5.7:** Example of ACE-SIS hourly carbon ion flux energy spectra reconstruction for extrapolation to higher energies on 02 March 2012, 00:00. Dashed lines correspond to the corresponding extrapolation method flux fitting results. Squares correspond to the ACE-SIS hourly carbon flux. The coloured lines are the existing carbon flux data with different CRÈME 96 SPE models. WW stands for the worst week, WD means the worst day, and P5M stands for the peak five minutes.

Since the ACE-SIS database only provides 14 types of heavy-ion flux data, it is essential to analyze the effect of limited types of heavy-ions for this study. Table 5.2 describes the target SRAM SEU rate estimated for all ions from He (2) to U (92) and the ACE-SIS detected ions by using the CREME96 SPE models, respectively. The results show that in all three CRÈME 96 SPE models, the error caused by the incompletely detected ion type is less than 1%. Therefore, only using ACE-SIS ions for the heavy-ion-induced upsets analysis is accepted. However, the energy range of the ACE-SIS is from 5 to 150 MeV, which is relatively low and insufficient for the following SEU rate estimations. In order to reconstruct a suitable energy spectrum for ACE-SIS ions with a proper energy range, i.e., from 1 MeV to 1 GeV, the first-order power-law fit [TDB<sup>+</sup>96] is used to extrapolate to a higher energy range. The power-law distribution appears as a straight line in the plot with a constant slope and tends to provide a spectrum with an excess of high- and low-energy particles. To get the hourly energy spectra for each of the target solar events examined in this thesis, the SPE ion energy spectra reconstruction procedure begins by picking the hourly ion flux from the ACE-SIS dataset. Because the ACE-SIS provides 14 ion flux data each hour, the reconstruction procedure is carried out using the hourly average ion flux in the eight energy channels of each ion type. In addition, the arithmetic moving average process, which calculates by using nearby ion flux and then dividing by the number in the calculation average, was applied to evaluate the hourly fluxes data when the source data is invalid at some moments. Furthermore, the SPE reconstructed energy spectra for all ions using the power-law fit to the three flux locations

with the greatest energy values [HED04]. For example, Figure 5.7 presents the spectra derived from the power-law fit of ACE-SIS Carbon ions' one-hour average flux data with respect to existing CREME96 SPE models, respectively. The reconstructed hourly spectrum for 08 March 2012, 00:00, which is before the peak of one large solar event.

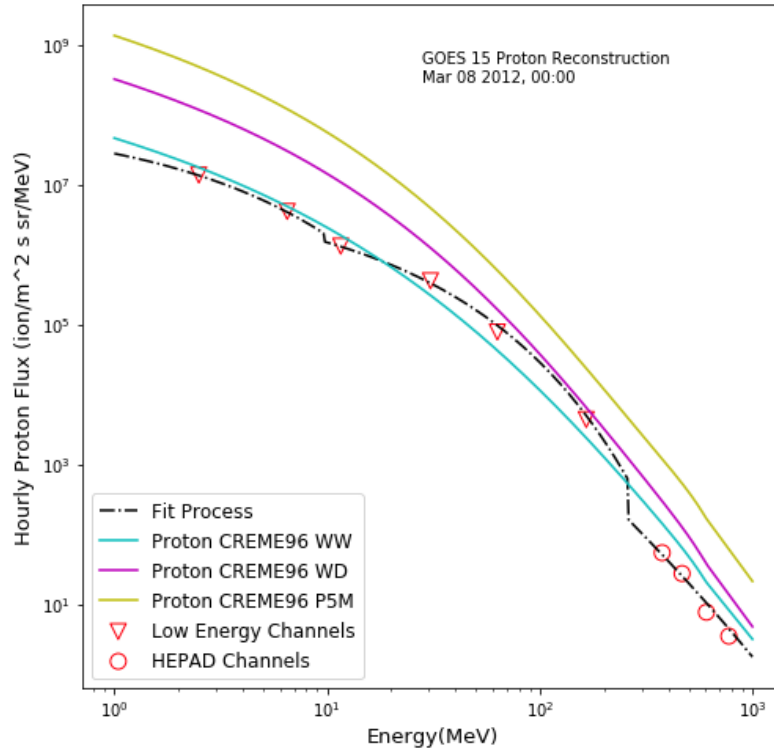
The proton data from the GOES database has been available in good quality and sufficient energy range (over 700 MeV) for the energy spectra reconstruction. This thesis applies the first-order exponential in rigidity approach for the proton data. SPE proton spectra are used to assess astronaut radiation exposure in space, either in integral (protons/cm<sup>2</sup>) or differential (protons/(cm<sup>2</sup>·MeV)). Various analytical and functional approaches have historically been used to fit satellite data. The first-order exponential in rigidity approach fitting method has been employed in the scientific community for several decades due to particle detectors' poor energy resolution capabilities in the early "Space Age" era. The integral energy spectrum's mathematical representation is:

$$\varphi(> E) = N_0 e^{-R/R_0} \quad (5.1)$$

where the  $\varphi(> E)$  is the integral energy fluence in proton/cm<sup>2</sup>,  $N_0$  is a normalization constant,  $R$  is the proton rigidity (proton momentum) in MV (million volts), and  $R_0$  is the characteristic rigidity in MV. The proton rigidity  $R$  is related to the proton energy (MeV) by:

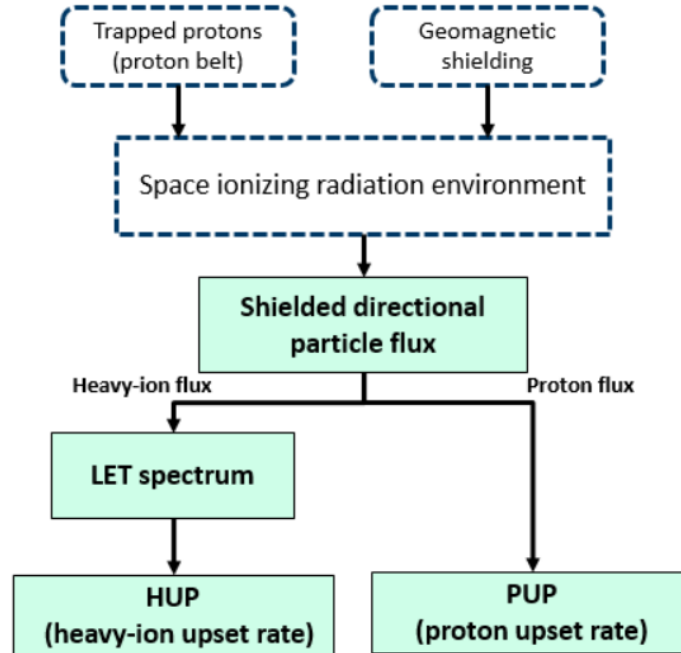
$$R(MV) = \sqrt{E^2 + 2m_0E} \quad (5.2)$$

where the  $E$  is the proton energy in MeV and  $m_0$  is the rest mass of the proton (938 MeV). Since the above approach fit performs not so well when the energy is very low, the power-law fit method is used when  $E$  is less than 10 MeV. Furthermore, in order to estimate the proton energy spectrum in the same energy range as the ACE-SIS heavy-ions, the proton data were extrapolated from the highest energy value by applying the power-law fit method to the GOES HEPAD proton data (highest energy channels, greater than 375 MeV for GOES 13 and 15). Furthermore, the GOES database provides 5-minute average flux proton data. The hourly energy and hourly average proton energy spectra, like the heavy-ion spectra, were derived from the accessible GOES energy channels. As an example, Figure 5.8 shows GOES 15 proton one-hour average flux data and the corresponding fitting process results obtained with respect to the CREME96 SPE models on 08 March 2012, 00:00, respectively.



**Figure 5.8:** Example of GOES hourly proton ion flux energy spectra reconstruction on 02 March 2012, 00:00. Dashed lines correspond to the corresponding extrapolation method flux fitting results. Triangles correspond to the hourly proton in the GOES low energy detector. Circles correspond to the hourly proton in the GOES HEPAD detector. WW stands for the worst week, WD means worst day, and P5M stands for peak five minutes.

Following the reconstruction operation, the hourly average flux spectra for all ACE-SIS ions and protons for all chosen targets were acquired. Furthermore, this technique has revealed two crucial issues regarding the reconstruction of solar event energy spectra: sufficient energy data and continuous particle data are necessary. The energy spectrum reconstruction methods introduced in this section (i.e. power-law and exponential in rigidity) are two of the most common and easy-to-use methods. There are a lot of other reconstruction methods, such as the Band fit function [JWW10], which is a power-law that rolls smoothly into a second with a functional form that is continuous in both its value and derivative, the Ellison-Ramaty spectral fit [ERR85] which has a power-law shape at low energies and with an exponential rollover at high energies, the Weibull distribution [MAX<sup>+</sup>00] function which has a similar scheme of exponential spectrum modelling, the Bessel function expression [MTR<sup>+</sup>84] which derived from stochastic acceleration arguments.



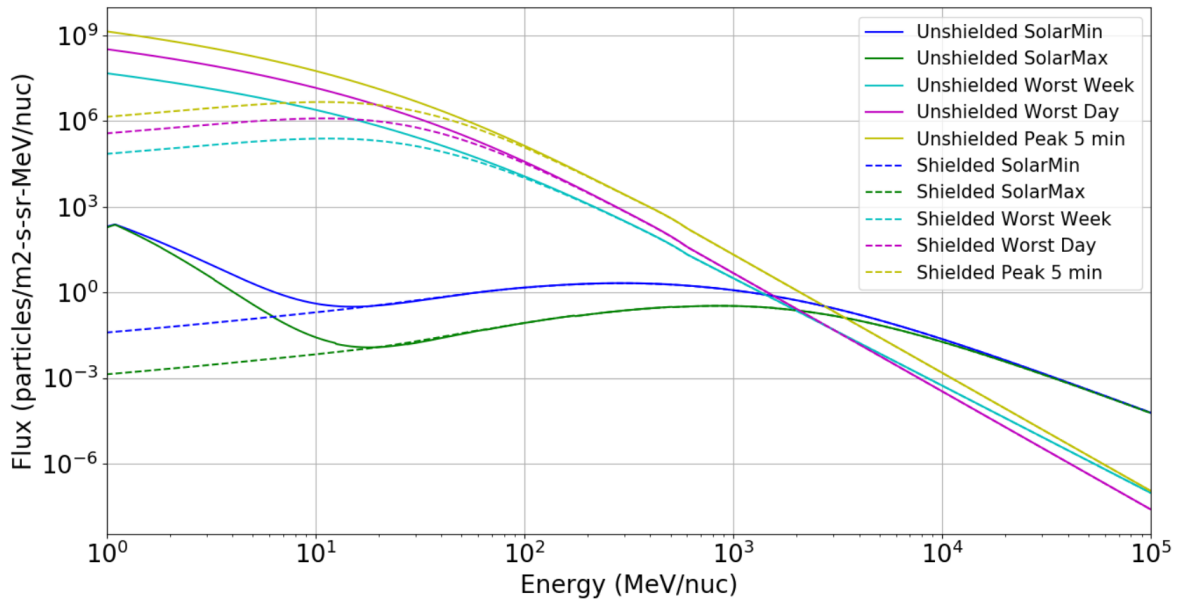
**Figure 5.9:** CREME96 flow chart for calculating heavy-ion and proton-induced upset rates. A dashed block indicates an optional calculation step. The green blocks represent the main focus of this chapter.

### 5.2.3 SEU Rate Estimation

By applying the reconstructed energy spectra and the target SRAM cross-section parameters in the CREME96 suite, the hourly heavy-ion and proton-induced SEU rate for the target SRAM during selected events can be obtained. Figure 5.9 presents the general flow chart of the CREME96 suit for the calculation of the heavy-ion and proton-induced SEU rate. It is essential to mention that the trapped proton calculation evaluates the fluxes of trapped protons for satellites inside Earth's magnetosphere. Moreover, the geomagnetic shielding calculation evaluates the shielding effect of the Earth's magnetic field, which is only necessary for space crafts below the Geosynchronous Equatorial Orbit (GEO). Since this thesis focuses on deep-space particles generated by solar events, the GOES and ACE satellite orbits are outside the Earth's magnetic influence. Thus, this thesis does not consider the trapped protons and geomagnetic shielding.

The CREME96 suit recommends using the space ionizing radiation environment model for spacecraft in geosynchronous orbits or orbits higher than geosynchronous orbits, which provides a numerical model of the space ionizing radiation environment on the spacecraft's surface and then transmits through shielding. The space ionizing radiation environment model is also the recommended starting point for highly elliptical orbits evaluation, with perigees at low altitudes and apogees at or beyond geosynchronous because most of the orbits would be exposed to interplanetary flux. This approach may be somewhat conservative because it ignores the shielding provided by the solid Earth in the low-altitude part of the orbit. In this thesis, the obtained hourly reconstructed heavy-ion and proton energy spectra fluxes, as introduced in Section 5.2.2, have

already performed the required ionizing radiation environment evaluation at the external surface of the spacecraft. Therefore, this model is ignored during the proposed SEU evaluation flow. The collected hourly reconstructed energy spectra files would be directly used as the unshielded directional hourly heavy-ion and proton fluxes, which is the output of the space ionizing radiation environment model.

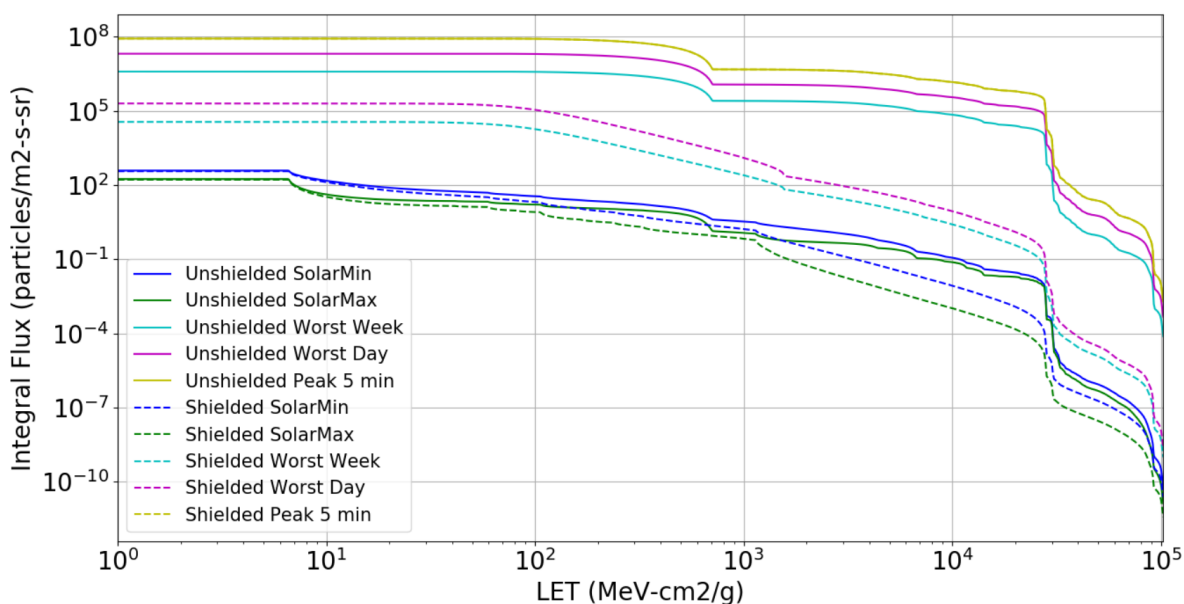


**Figure 5.10:** Example of unshielded and shielded (100 mils of aluminium) directional proton flux in differential form as a function of the energy for the five existing CRÈME 96 space environment modules.

The shielded directional particle flux model can be determined by transporting the nuclear fluxes through the spacecraft shielding. This module takes into account both energy loss and nuclear fragmentation, which provides the shielded directional particle flux under the determined shielding parameter. For spacecraft, it is necessary to use shielding material, such as aluminium, to prevent the encountered of single-event effect from low-energy particles. Since no specific space projects are focused on in this thesis, 100 mils of aluminium shielding are assumed, which is the conventional equivalent shielding thickness for spacecraft [VGU17] and the default value in the CRÈME 96 suit. It is worth mentioning that when the actual shielding layer is much thicker, using the nominal shielding layer thickness may cause a severe overestimation of the radiation environment. Average or nominal shielding thickness is rarely sufficient for reliable SEU rate calculations during solar events. On the contrary, careful handling of shielding is significant because the solar spectrum usually drops sharply with increasing energy. In these cases, using only a single shield thickness can easily lead to a tenfold error in the predicted SEU rate. The SEU calculation should be performed using the true distribution of shielding thickness for the solar particle environment. Figure 5.10 presents an example of unshielded and shielded directional

proton flux as a function of the energy for five existing CRÈME 96 space environment models. It illustrates that the flux with shielding would be significantly lower than the unshielded peer when the corresponding energy is low. However, the shielded directional flux converges to the unshielded directional flux when the energy is high. In this thesis, all of the collected hourly reconstructed energy spectra files must be processed by this module and obtain the corresponding shielded hourly reconstructed energy spectra.

After collecting the shield particle fluxes, the LET spectra module is necessary for heavy-ions. This module intends to consider different energetic ions together by representing the particle environment as the summation of the fluxes of all energetic particles as a function of their LETs, allowing the LET to express the energy transfer of many different ions in one variable. Moreover, the LET spectrum is required as an input for calculating the rates of SEEs due to direct-ionization. Figure 5.11 presents an example of the unshielded and shielded directional heavy-ion (He(2)-U(92)) flux in the integral form as a function of the LET (ions that exceed the specified LET) for five existing CRÈME 96 space environment models. The shielded LET spectrum is calculated by bypassing the shielded directional particle flux block, as shown in Figure 5.9. It illustrates that the shielded values are significantly lower than the unshielded peers. In this thesis, all of the shielded hourly reconstructed energy spectra of heavy-ion fluxes must be processed by the LET spectra module. As the output, the hourly LET spectrums can be obtained and used for hourly heavy-ion induced SEU rate calculation.



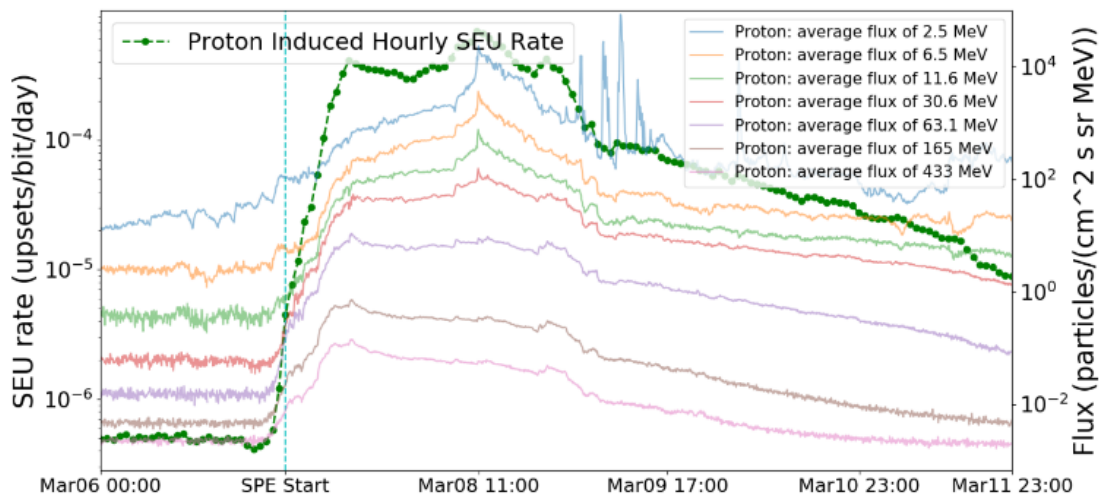
**Figure 5.11:** Example of unshielded and shielded (100 mils of aluminium) directional heavy-ion flux in the integral form as a function of the LET for the five existing CRÈME 96 space environment modules.



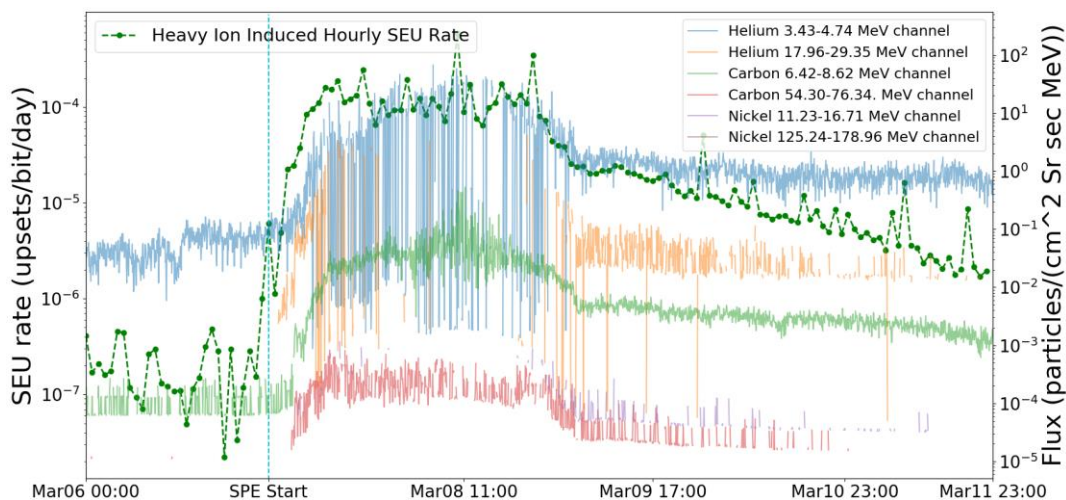
The CREME96 suit's Proton-Induced SEE Rate Calculation (PUP) module calculates the proton-induced upset rate by estimating the single-event effect rate owing to proton-induced nuclear recoils. It is calculated by integrating the Weibull distribution of the cross-section parameters (example shown in Figure 5.6) and shielded directional proton flux (example shown in Figure 5.10). Moreover, the proton cross-section needs to be specified in units of  $10^{-12}$  cm<sup>2</sup>/bit vs proton energy in MeV. In this thesis, this module needs to process all of the collected hourly reconstructed proton energy spectra files. As the output of a single process of one hourly reconstructed proton energy spectra file in the PUP module, the hourly proton-induced SEU rate of the target SRAM in the unit of upsets/bit/day can be obtained. Figure 5.12 presents the obtained proton-induced hourly SEU rate regarding the corresponding ion flux from 06 March to 11, 2012. Due to the high quality of the proton data in the GOES database, the quality of the calculated proton-induced hourly SEU data also corresponds well.

The Heavy Ion-Induced Direct Ionization-Induced SEE Rate Calculation (HUP) module computes the single-event effect rate owing to heavy-ion-induced direct ionization. As introduced in Section 2.2, the heavy-ion-induced SEUs depend on the energy deposition, but not the number of hits like proton-induced SEUs. Thus, the Sensitive Volume (SV) geometry is needed for the heavy-ion induced SEU rate estimation. The upset rate due to direct ionization is computed in the HUP module using the Rectangular Parallelepiped (RPP) model [EPE11], in which the bit-sensitive volume is considered to have this shape. To use this approach, the sensitive volume dimensions (x,y,z) in microns must be given. Moreover, since the SRAM cross-section with Weibull distribution parameters is intended to be applied in the HUP module, as shown in Figure 5.6, the integral RPP method, in which the upset rate is calculated from a numerical integration over the LET-dependent cross-section, is used. It is not easy to select an RPP thickness that complies with the device's cross-section direction dependency. RPP dimensions can be acquired from chip designers and manufacturers in some situations. The size of the RPP can also be estimated by carefully analyzing accelerator ground-test data [PPS<sup>+</sup>93]. In this thesis, since the RPP dimension of the target SRAM is not available, the rules of thumb provided by CREME96 are used to estimate a reasonable value: if the SEE cross-section measurements from the accelerator reveal a limiting value, assume the sensitive volume has a square surface area and set  $x = y =$  square root of the limiting cross-section per bit; The assumed "z" value should be compatible with the technology, and common charge collecting depths for SOS and SOI are 0.5 microns and 2 microns for CMOS/epi [PPA<sup>+</sup>92]. Furthermore, the "z" dimension is the lowest RPP dimension in general. In accelerator SEE cross-section measurements, "z" is defined as the device's thickness along the beam under typically incident irradiation. Because "z" provides the distance scale in translating from effective LET to deposited charge, SEE rate estimates can be very sensitive to it. Applying the above RPP parameters, the Weibull distribution of the target SRAM's cross-section parameters and the shielded directional heavy-ion flux in the integral form in the HUP module, the hourly average heavy-ion-induced SEU rate in the unit of upsets/bit/day can be collected. Figure

5.13 presents the obtained ACE-SIS ions-induced hourly SEU rate regarding the corresponding ion flux from 06 March to 11, 2012. Due to the data gaps in the ACE-SIS database and the missing flux data, there is a sizeable upset fluctuation of the SEU rate.



**Figure 5.12:** March 6-11, 2012 SRAM hourly SEU rate estimated from GOES proton database. The particle flux for all lower to higher energy channels is shown, and all channel data are of good quality.



**Figure 5.13:** March 6-11, 2012, SRAM hourly SEU rate estimated from ACE-SIS heavy-ion database. The particle ion flux of He, C and Ni for the lower and higher energy channels is shown, and the data is of poor quality.

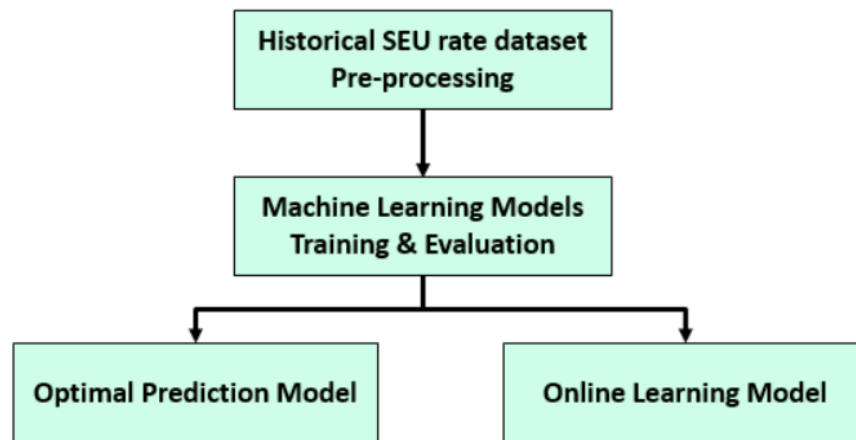
The last step is to determine the final hourly SEU rate of the target SRAM during the selected solar events by adding up both the proton and heavy-ion induced hourly upset rate. In addition, the final hourly SEU rates for target SRAM during the selected events are the sum of calculated proton and heavy-ion induced SEU rates. According to Tables 4.4 and 4.5 (i.e., target SRAM CREME96 SEU rate table), the proton-induced SEU rates are supposed to be larger than the heavy-ion-induced SEU rate during the SPE period as the shown example (Figure 5.12 and 5.13). Therefore,

the upset fluctuations of the heavy-ion-induced SEU rates are acceptable for this study. By evaluating all of the collected hourly heavy-ion and proton flux data from historical solar events, the SEU rate dataset of the target SRAM-based monitor can be obtained.

### 5.3 SEU Prediction with Machine Learning

This section elaborates on the model training process (as shown in Figure 5.2), which applies machine learning techniques for the early detection of SEU rate changes. Figure 5.14 presents the evaluation flow of the proposed model training process. The collected hourly SEU rate dataset from analyzing historical solar events, as introduced in Section 5.2, is used in this section as input for machine learning training. As the output of this process, two machine learning models can be selected. One is used to predict changes in hourly SEU rate, thus, detecting the spatial environment early, and the other is an online learning model for optimizing the previous prediction model parameters during run-time.

The hourly SEU prediction in this thesis is intended to operate in conjunction with real-time in-flight SEU measurement, and the approach aims to predict fine-grained SEU rates in advance by using the upset rates of the  $n_h$  last hours provided from the SEU monitor. The time-series regression approach is appropriate for the forecast aim. It is a statistical strategy for forecasting future responses based on past responses and the dynamic transmission of relevant predictor variables. Furthermore, the time-series approach, which is extensively used for modelling and predicting economic, financial, and biological systems, may help users understand and anticipate the behaviour of dynamic systems using experimental or observational data.



**Figure 5.14:** Evaluation process of the model training process for collecting, selecting and training machine learning models for space environment prediction.

In this thesis, several existing machine learning regression models with the time series method have been trained and evaluated to select the optimal model, yielding the best prediction accuracy. The regression models were selected based on a low-resource demand. The model training was

conducted offline and supervised by applying the estimated hourly SEU dataset from past solar events. An already trained model can be used online to perform the prediction, which generally needs fewer computation resources. Therefore, several machine learning regression models have been trained and evaluated to select the optimal model, yielding the best prediction accuracy. In addition, an online learning method is evaluated to increase the prediction accuracy of the selected regression model during the in-flight online running phase, which could let the regression mode adapt to a changing working environment.

To train the machine learning models, the in-flight hourly SEU rate data received from previous solar occurrences is first processed and translated to reflect actual SEU monitor disturbances. Following that, the machine learning regression model is trained using converted data from historical solar events. Then, the accuracy of the trained model in predicting the hourly upset rate is evaluated. Finally, an online learning method analyses the selected model parameters optimization. These steps are described in detail in the following subsections.

**Table 5.3:** Example of the pre-processed SEU rate dataset around the start point (highlight in green) of one solar event on 07 March 2012. "Upset rate" is in unit SEU/bit/day, "Upset" is in unit SEU/2Gbit/hour the normalized upset is the min-max scaling of the hourly upset data;  $n_h$  last hours (i.e. three last hours in this example, highlighted in grey) are the normalized upsets rate collected from previous hourly upset rates.

Time	Upset Rate	Upset	Normalized Upset (0h)	-1h	-2h	-3h
2012-03-07 03:00:00	$2.93 \times 10^{-6}$	262	0.001999	0.000465	0.000394	0.000999
2012-03-07 04:00:00	$4.51 \times 10^{-6}$	403	0.003075	0.001999	0.000465	0.000394
2012-03-07 05:00:00	$7.14 \times 10^{-6}$	643	0.004906	0.003075	0.001999	0.000465
2012-03-07 06:00:00	$1.02 \times 10^{-5}$	919	0.007011	0.004906	0.003075	0.001999
2012-03-07 07:00:00	$2.36 \times 10^{-5}$	2123	0.016197	0.007011	0.004906	0.003075
2012-03-07 08:00:00	$3.88 \times 10^{-5}$	3496	0.026672	0.016197	0.007011	0.004906
2012-03-07 09:00:00	$6.36 \times 10^{-5}$	5726	0.043686	0.026672	0.016197	0.007011
2012-03-07 10:00:00	$1.11 \times 10^{-4}$	9946	0.075882	0.043686	0.026672	0.016197

### 5.3.1 Pre-processing of the Dataset

The in-flight hourly SEU rate data acquired from historical solar events form the test and training dataset for the machine learning models. The hourly SEU rate values are obtained by processing the hourly flux database, as explained in Section 5.2. The processed hourly flux data, together with the cross-section of the target SRAM, are used to calculate the hourly SEU rate by using the CREME96 suite. In this way, the SEU/bit/day is obtained. Since, in the actual system, the data from the SEU monitor can be an integer ranging from 0 to the size of the SRAM, the calculated SEU/bit/day is multiplied by the target SRAM size. For example, the SRAM size of 2G bit is

selected, which can provide enough detected SEU resolution during any solar events. Further, to get the hourly upset rate, the SEU/bit/day values are divided by 24h. Thus, the SEU rate per hour is obtained as measured from the SEU monitor in the actual system.

Most machine learning models do not perform well, when the input data has a wide numerical range. Therefore, a min-max scaling was applied to the data before the training, which scales the input data from 0 to 1. This is achieved by dividing the hourly SEU rate, obtained from the previous step, by the next power of two of the highest expected number of upsets. In this way, no actual division needs to be implemented in hardware since it is just a different representation of the input data as a fixed point integer. As an example, a part of the pre-processed SEU rate dataset around the start point of one sizeable solar event on 07 March 2012 is shown in Table 5.3 with the  $n_h$  last hours in 3. It is essential to mention that the actual pre-processed table would have a much larger  $n_h$  last hours range to 24. Since the time series regression method is intended to use for predicting the following SEU rate, thus, the  $n_h$  last hour part in the pre-processed table (i.e. grey part in Table 5.3) would be analyzed by a series of machine learning models to predict the hourly normalized upset data.

### 5.3.2 Model Training

The transformed and pre-processed data is used to train and evaluate different machine learning regression models in a supervised manner. The data set was split, where 60% of the data was used to train the model and the remaining 40% of the data was used to evaluate the model. In this study, five well-known regression models have been analyzed: (1) Linear Regression, (2) Decision Tree Regression, (3) k-Nearest Neighbors Regression, (4) Multi-Layer Perceptron (MLP) Neural Network and (5) Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). Python's Scikit-Learn [FPE<sup>+</sup>11] and Keras [FCH<sup>+</sup>15] frameworks are used to implement the above models. Scikit-learn (Sklearn) is one of Python's most useful and robust machine learning packages. It offers a set of fast tools for machine learning and statistical modeling, such as classification, regression, clustering, and dimensionality reduction, via a Python interface. This Python-primarily created package is based on NumPy, SciPy, and Matplotlib. Aside from that, Keras is one of the most influential and user-friendly Python libraries, with a basic structure that gives a clean and simple approach to generating deep learning models based on TensorFlow or Theano. Keras is intended to let users easily define deep learning models. The investigated regression models are briefly described as follows:

- 1) Linear Least Squares Regression: The linear least-squares algorithm is based on the standard linear model. It is a set of formulations for solving statistical problems involved in linear regression, including variants for ordinary (unweighted), weighted, and generalized (correlated) residuals. The target output variable is represented as a linear combination of the input variables. The algorithm aims to minimize the squared sum of

the difference between the true value in the training dataset and the predicted value by linear approximation.

- 2) **K-Nearest Neighbors Regression:** The k-Nearest Neighbor algorithm uses similarity in the input variables to predict values of new data points. The training data set is only indexed and stored in a database during the training phase. A new data point is predicted based on how closely it corresponds to the points in the training set. A weighted average of the k-nearest neighbours is used to predict the value. The main hyperparameter of the model is k, the number of nearest neighbours used for the prediction.
- 3) **Decision Tree Regression:** Decision Tree models recursively partition the input feature space by inferring simple decision rules from the training data set. A tree structure represents the data. The tree branches represent the decision rules, and the leaves contain the trained values. The primary hyperparameters for this model are used to control the tree's structure, such as the maximum depth, the maximum number of leaf nodes and the balance of the tree. During the hyperparameter optimization, the model performs best when the tree structure is not constrained.
- 4) **Multilayer Perceptron Neural Network:** A MLP belongs to the feedforward Artificial Neural Networks (ANN) class. A MLP Neural Network consists of at least three layers of nodes. The first layer is the input layer, followed by one or more hidden layers and an output layer. Except for the input nodes, each node is a perceptron. A single perceptron has one or more inputs, a bias, an activation function, and one output. The received input is multiplied by a weight and passed to the activation function, which produces the output. The primary hyperparameters for this model are the activation function and the network topology.
- 5) **Recurrent Neural Network with Long Short-Term Memory:** The Recurrent Neural Network (RNN) is a generalization of feedforward ANN and is extended by internal memory. This means that the output of the current input depends on past computations. In the other considered models, all the inputs are independent of each other, but with the help of the RNN's memory, all the inputs are related to each other. The Long Short-Term Memory (LSTM) is a special kind of RNN, to overcome some of its issues with learning dependencies over longer sequences. This means RNN with LSTM is designed to process input data sequences, which makes them applicable for time series data. The main hyperparameter of this model is, similar to the MLP neural network, the network topology.

Usually, machine learning models have internal parameters or use internal states to adjust their algorithm and perform an accurate prediction. These parameters are determined and optimized based on the training data during the training process. In addition to the internal parameters, most machine learning models also have external parameters called hyperparameters. These hyperparameters are often used to tune the training algorithm, determining the internal parameters.

Thus, contrary to internal parameters, hyperparameters cannot be determined by the training algorithm itself. They need to be specified manually by the user before the training process. In order to find the optimal set of hyperparameters of a model, a hyperparameter optimization has to be performed. Therefore, the model is trained and evaluated several times with different sets of hyperparameters.

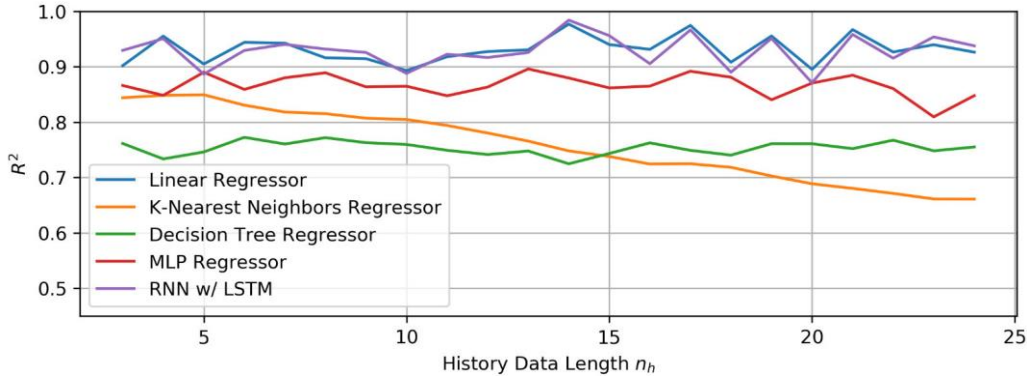
The hyperparameters are usually determined by performing a random search coupled with the grid search method [JBE<sup>+</sup>12], which performs hyperparameter tuning to determine the optimal values for a given model. In this thesis, the GridSearchCV function is used, which is a part of the Scikit-learn package. The models are first evaluated with randomly generated hyperparameter values in this approach. Then, a more detailed grid search is performed within the region of the best hyperparameter values obtained by the random search.

Besides the hyperparameters, the model's performance also depends on the number of past hourly SEU rate values used for the prediction. The length  $n_h$  describes how many hours of past SEU rate values are used. Similar to the hyperparameter optimization, the optimal length for  $n_h$  needs to be determined. Thus, models were also evaluated for different history lengths of the hourly SEU data.

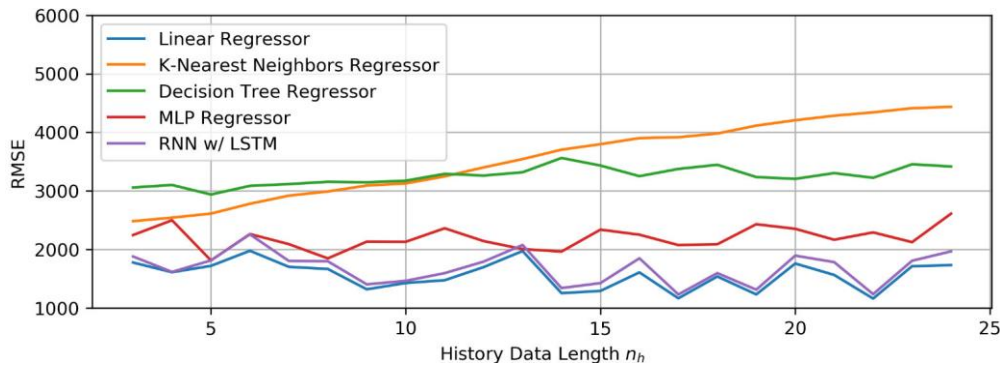
### 5.3.3 Model Evaluation

To evaluate the prediction performance of a model, in this thesis, the root-mean-square error (RMSE) and the coefficient of determination ( $R^2$ ) metrics are mainly used. The RMSE describes the square root of the quadratic error of the expected values. Compared to the mean absolute error, the root-mean-square error gives more weight to more significant errors that are then penalized more. The  $R^2$  score is a combined evaluation metric which takes the RMSE and the variation (dispersion) of the model into account. Thus, the  $R^2$  score provides a measure of how well future samples are likely to be predicted by the model. These metrics were calculated by comparing the test data set with the predicted data set of the trained model. In addition, the Mean Absolute Error (MAE), which measures errors between paired observations expressing the same phenomenon, and Maximum Absolute Error (MAX), which performs the difference between the actual and measured value, are also used as references in the evaluation process.

Cross-validation, a resampling strategy that employs various chunks of the data to test and train a model on different iterations, is used to produce a more reliable measurement. It is mostly utilized when the aim is known and one wishes to assess how well a predictive model can perform in practice. This thesis uses the cross-validation strategy with a cross-validation fold of 10. In this strategy, the dataset is split into ten different trains, and test data sets are used to train and evaluate the models independently. Then, the calculated metrics for each independently trained and evaluated model are averaged over the different measurements.



**Figure 5.15:**  $R^2$  scores (higher the better) for the selected regression models with varying history data length  $n_h$ .



**Figure 5.16:**  $RMSE$  scores (lower the better) for the selected regression models with varying history data length  $n_h$ .

The performance of the models was evaluated for different length  $n_h$  of the past hourly SEU data, considering the hourly intervals between  $3h$  and  $24h$ . For each considered  $n_h$ , the above-described hyperparameter optimization was performed, and the model performance was measured according to the specified metrics. Figures 5.15 and 5.16 show the  $R^2$  scores and  $RMSE$  for each regression model on different history data lengths, respectively. It can be seen that both the RNN with LSTM and the linear regression models have good accuracy, and the performance of the RNN is slightly better. The best performances are obtained with past hourly SEU data  $n_h$  of 14 for the RNN and 17 for the linear regression model.

Given that RNNs are a type of machine learning method that is commonly used for time-series and sequential data applications, it is not unexpected that the RNN with LSTM has the best prediction performance in the selected models. RNN, on the other hand, necessitates a high computing capability and a wide memory space, and the related hardware expense on the embedded system could not be accepted [RPN<sup>+</sup>20]. Figure 5.17 depicts the LSTM layers in the RNN network, where  $x_t$  represents a vector combination of the LSTM layer input,  $h_{t-1}$  denotes the previous time-step output vector, and  $h_t$  is the output vector at time  $t$ . An LSTM has a memory cell state  $C_t$  and three gates, according to its construction:



- Forget gate: it is used to determine which information should be forgotten and is computed as follows:

$$f_t = \delta(W_f[h_{t-1}, x_t] + b_f) \quad (5.1)$$

where  $W_f$  is the weight matrix,  $b_f$  is the bias vector, and  $\delta$  is the sigmoid function.

- Input gate: it determines which information is to be refreshed and is computed in the same way as the forget gate:

$$i_t = \delta(W_i[h_{t-1}, x_t] + b_i) \quad (5.2)$$

where  $W_i$  and  $b_i$  are the weight matrix and bias vector, respectively.

- State computation: this model is used to compute the LSTM cell's new memory state  $C_t$ . It should first compute the potential values for the new state:

$$\hat{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (5.3)$$

where  $W_c$  and  $b_c$  are the weight matrix and bias vector, respectively. Then, the new state vector,  $C_t$  is calculated by the addition of the previous state vector  $C_{t-1}$  element-wise multiplied with the forget gate output vector  $f_t$  and the new state candidate vector  $\hat{C}_t$  element-wise multiplied with the input gate output vector  $i_t$  as:

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \quad (5.4)$$

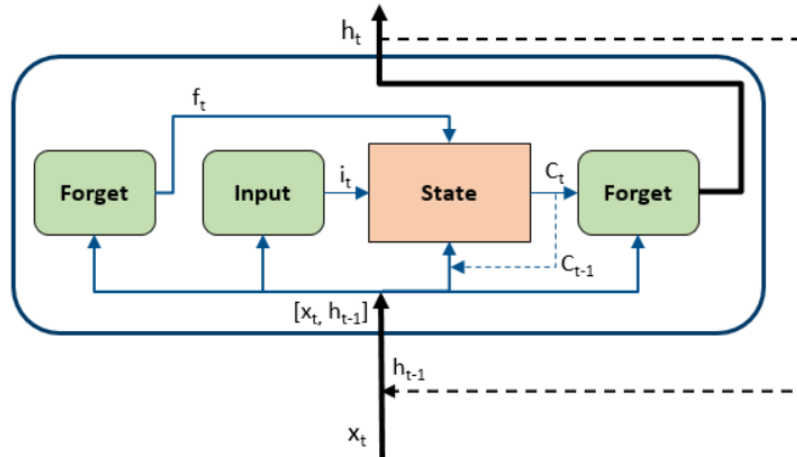
where  $\odot$  denotes the element-wise multiplication.

- Output gate: this module is to compute the LSTM cell output as:

$$h_t = \delta(W_o[h_{t-1}, x_t] + b_o) \odot \tanh(C_t) \quad (5.5)$$

where  $W_o$  is the weight matrix,  $b_o$  is the bias vector, and  $\sigma$  is the sigmoid function.

Matrix-to-vector multiplications dominate the number of computations and parameters required to perform the aforementioned LSTM cell calculations in an embedded system. For each matrix to vector multiplication, assuming the input vector  $x_t$  has the size of  $m$  and the  $h_{t-1}$  vector has the size of  $n$ , thus the multiplied with weight matrices has the size of  $(m+n) * n$ , which requires  $(m+n)n$  MAC operations and in a total of  $2((mn) + n^2)$  multiplications and additions. Because this computation is done four times within the LSTM computation, the total number of operations and parameters for an LSTM is multiplied by four. Therefore, the computational complexity of the RNN with LSTM model is very high and requires plenty of resources for computation and memory space, which may not be suitable for embedded design.



**Figure 5.17:** Block diagram of the recurrent neural network with long short term memory layer.

Although the prediction performance of the linear regression model is slightly worse than the RNN with LSTM model, the linear regression model is much simpler and requires significantly fewer resources [VSZ<sup>+</sup>17], which would be suitable for the low-cost embedded system implementation. Hence, the linear regression model has been chosen as the best option in this thesis for the hardware accelerator for the embedded system. The details of the linear regression algorithm and the corresponding hardware implementation are detailed in Section 5.4.

It is worth mentioning that the current prediction for one hour in advance is a case study, which is consistent with the recommended fault detection period of the target SRAM-based SEU monitor and the resolution of historical space flux data from public databases (as shown in Figure 5.7 and 5.8, the hourly average flux data are used for energy spectra reconstruction). Estimation and verification of the prediction several hours in advance will be addressed in future work, and for that purpose, additional in-flight data could be required. Moreover, the dependable system would not operate without any protection, even at low radiation levels. The system's fault-tolerance manners provide enhanced protection at high radiation levels. However, in practice, a single particle could hit a sensitive node in the circuit (and not in the monitor), causing a failure. Thus, critical elements need to be protected to a certain extent all the time. For example, clock and reset trees may be the source of multiple errors, which must be avoided by applying static design measures at low design levels (e.g., gate upsizing and SET filtering). Similarly, the memory elements storing critical data must be permanently protected with local redundant mechanisms. Then, at the system level, enhanced protection is activated as required to reduce the overall soft error rate under highly critical radiation intensity. Therefore, with carefully applied selective fault-tolerance mechanisms, the system's failure can be avoided in the first instant of a particle strike, and the proposed prediction of one hour in advance would not be too late to adapt to the change in the environment.

Moreover, due to the fact that an SPE phenomenon is an explosive event, the change in space radiation intensity is abrupt. Therefore, the first SEU instance after the event explosive cannot be accurately predicted only from the detected background SEU data. However, the missing prediction of the first instance would not put the dependable system at risk. Since comparing with the background flux, the flux during the first instance flux is only a few times higher, which is much lower than the SPE peak flux. According to the analysis of all the selected SPEs in the solar cycle 24, the target SRAM SEU rates for the first detected instance after solar event explosive is usually 1.5~6 times higher than the background condition, which is usually less than the recommended self-adaptive mode trigger thresholds, which is introduced in Chapter 6. In addition, the dependable system would not operate without any protection, even at low radiation levels. After detecting the first instance after the event explosive, the proposed one-hour in advance prediction allows to continuously and accurately estimate the forthcoming SEUs. In addition, after the detection of the first instance, an appropriate radiation protection mode can be quickly deployed if necessary (do not have to wait for the prediction data), and the proposed one-hour in advance prediction allows to continuously and accurately estimate the forthcoming rise or fall in SEU rate before that information is available from the monitor. Therefore, the general dependable system in space application can tolerate the first instance particle strike, and the proposed prediction method is suitable for timely adapting the system to the changing environment before the onset of high radiation levels.

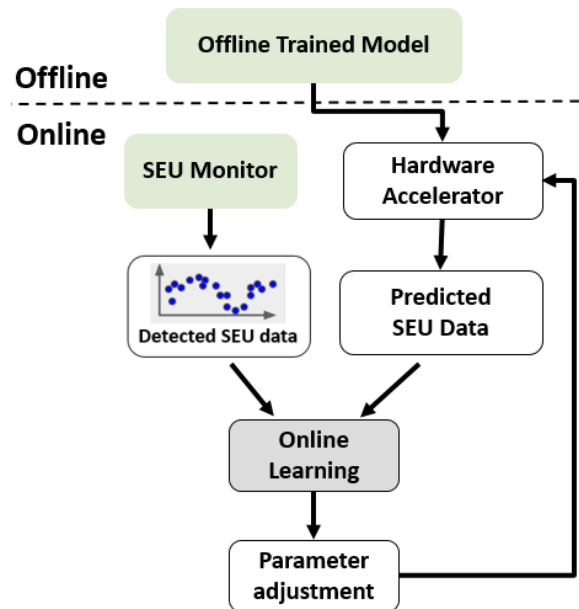
#### **5.3.4 Online Parameter Adjustment**

Since the offline trained machine learning model is obtained according to the historical ion flux data, the proposed model may not perfectly fit the real working environment and could not adapt to a changing environment. The primary motivation for combining online adjustment and offline prediction is to enable the adaptation to completely unexpected situations. Although the SPEs are the main concern for satellite missions, deep-space missions may also encounter a more significant number of unpredictable bursts of galactic cosmic rays. The information on such scenarios may not be publicly available, and thus the offline training would not be able to predict all possible conditions. In addition, offline training is based on a limited data set and certain ideal parameter assumptions. Thus, in the actual application, many other possible reasons would affect the accuracy of the offline algorithm, such as:

- Actual shielding parameter. This thesis assumes 100 mils of aluminium shielding, which is the conventional equivalent shielding thickness for spacecraft. However, the actual shielding parameters may be hard to know, and the monitor located in the satellite may also influence the shielding behaviour. For example, if the monitor is placed on the satellite side facing away from the Sun, the equivalent shielding could be much bigger than the assumption shielding parameter.

- Changing of the satellite orbits. The current offline SEU data set, calculated by the CREME96 suit, is analyzed for the near-earth interplanetary and the geosynchronous orbit satellites. Thus, the geomagnetically trapped protons and the shield protection from the Earth's magnetic field, which are essential for the upset analysis in low-Earth orbit satellites, cannot be neglected in the offline data collection process.
- Solar cycle changes. The current offline training data is based on the solar cycle 24 (2008-2019), which is the most recently completed solar cycle and the maximum value was substantially lower than other recent solar cycles, down to a level which had not been seen since solar cycles 12 to 15 (1878-1923). Therefore, the collected prediction model from this cycle may not be perfectly applicable to other solar cycles.

For these reasons, the online adjustment method is essential to ensure the reliability of the detector's response in a long-term mission, and it is necessary for improving the collected offline trained model through learning from the data obtained in the actual working environment. Online learning is suitable for systems that receive data as a continuous flow and is based on learning autonomously and incrementally from a stream of incoming data. Each learning step is fast and cheap compared to offline training. Thus, the system can learn about the new data on the fly.

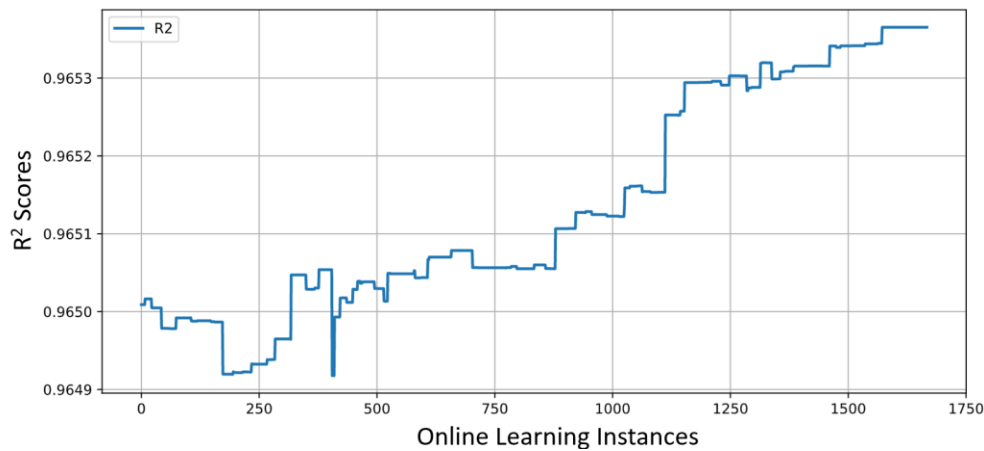


**Figure 5.18:** Block diagram of the proposed online SRAM SEU rate prediction function parameter adjustment procedure.

In Figure 5.18, the online learning process is illustrated. The SEU monitor can perform real-time SEU data detection during the online phase, as shown in Figure 5.2. Therefore, the online system can be trained incrementally by feeding the detected SEU data from the monitor sequentially and predicted SEU data from the prediction model. Thus, the offline trained prediction

model parameters can be continuously optimized in real-time, thereby adapting to the changing operating environment. In this thesis, the widely used online learning algorithm, Stochastic Gradient Descent (SGD) [QCU<sup>+</sup>19], which is an iterative method for optimizing an objective function with suitable smoothness properties, is used to update the parameters of the selected linear regression model. The SGD performs the gradient descent on a single instance for each training step, which is the predicted SEU and the corresponding detected SEU data pair. The SGD algorithm makes the model optimization process fast because it operates on a small amount of data at a time. Moreover, only one training instance of data needs to be stored in memory at each iteration, which is one of the main reasons for selecting the SGD algorithm.

In order to evaluate the SGD online learning performance, the data set used in the offline training was reused here. The data set was randomly split, where 50% was used to get the initial linear regression offline training model, 30% of the data was used for online learning training, and the remaining 20% of the data was used for online learning performance validation. The initial linear regression model parameters are the cornerstone of further optimization. The SGD algorithm optimizes the initial model based on this study's online learning training data set, 1668 instances. The algorithm detail and implementation of SGD in this study are described in detail in Section 5.4.2. After the SGD performs computing on a single instance each time, the new parameters would be generated and updated to the linear regression SEU prediction model. After learning about each instance, the evaluation is performed in the same approach described in Section 5.3.3.

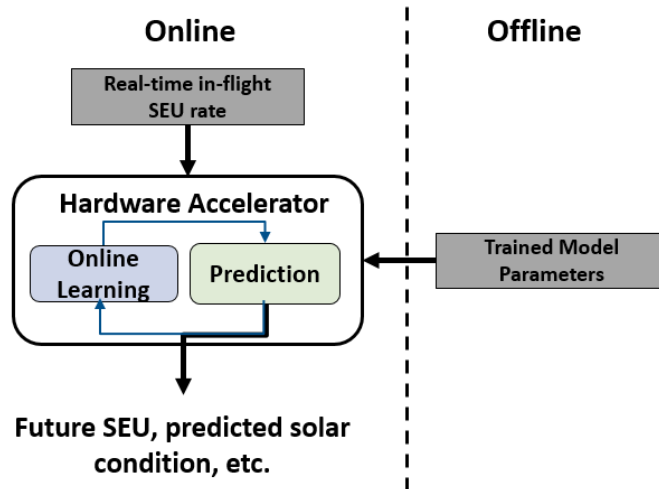


**Figure 5.19:**  $R^2$  scores for the online parameter adjustment evaluation on the dataset.

Since only one instance at each online learning step is used, the SGD algorithm is much less regular than the other gradient descent approaches. The cost function of the SGD always bounces up and down and decreases only on average. Figure 5.19 presents the performance in terms of the  $R^2$  score for the SGD training instances on the test data set. It can be noticed that as more instances are processed, despite the evaluation results scores are bounce up and down, the overall prediction

performance has been improved. On the other hand, another critical parameter for SGD online learning is the learning rate, which decides the step size at each training step when calculating the descent. If the setting of the learning rate is too small, the optimization process would be slow, and if you choose too large, the optimization process would oscillate or even fail to optimize. A small learning rate is selected in this thesis, making a slight variation of scores for each instance. The detail of selecting the suitable learning rate in this study is described in Section 5.5.2.

The above evaluation shows that the online learning algorithm can gradually improve prediction accuracy when the online working environment is consistent with the offline analysis environment. In Section 5.5.2, the online learning application of the other two scenarios is analyzed in detail: when offline flux data are not available, applying online learning forms a prediction function from scratch; when the online working environment is not the same as the offline analysis hypothesis, using the online learning improves the offline prediction model.



**Figure 5.20:** Evaluation process of the proposed hardware accelerator model for performing the real-time space environment prediction and online prediction adjustment.

## 5.4 Hardware Accelerator Implementation

The hardware accelerator implements the machine learning algorithm based on the linear regression model and the online learning algorithm based on the SGD. Figure 5.20 presents the evaluation process of the proposed design, which is intended to collaborate with the real-time SEU rate from the SRAM-based SEU monitor and the offline-trained results from the selected linear regression model. Figure 5.21 shows the architecture of the proposed hardware accelerator design and the connection with collaboration models. Two register files record the detection of real-time hourly SEU data from the monitor and the linear regression training parameters for the SEU data prediction. An accumulator is used to implement the required calculation. The control logic selects

the inputs and the functionality of the accumulator as well as the updating of the parameter register file. To obtain the actual predicted SEU data, the right shifter processes the pro-extended data, such as the pre-calculated SEU data from the accumulator.

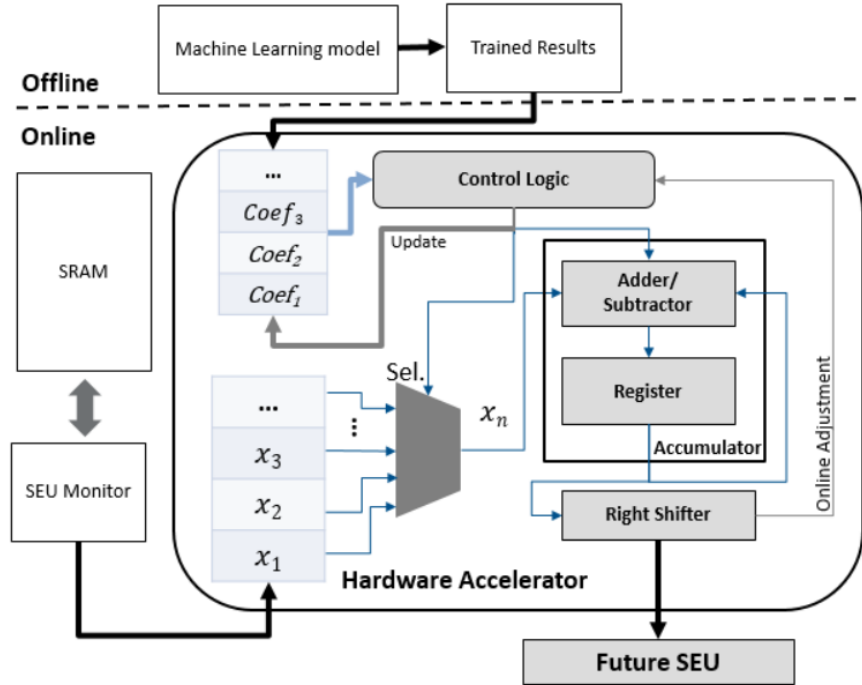


Figure 5.21: Proposed hardware accelerator design with the interface to external logic.

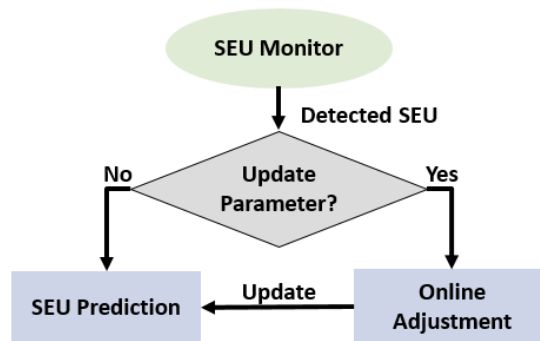


Figure 5.22: Decision flowchart of the proposed hardware accelerator.

The calculation flowchart of the hardware accelerator is presented in Figure 5.22. It contains two main stages: SEU prediction and the online model parameter adjustment. The two stages are used to implement the selected linear regression module and the SGD algorithm and are explained in the following subsections, respectively.

### 5.4.1 Single Event Upset Prediction

Based on the results from Section 5.3, the best accuracy of the SEU prediction can be obtained when the history data length of 17 is used. Moreover, the corresponding prediction function for the target SRAM, in this case, can be expressed as:

$$SEU_{pred\_acc} = 1.2929 * x_1 + 0.0868 * x_2 + (-1.1946) * x_3 + 1.0308 * x_4 + 0.1016 * x_5 + (-0.9142) * x_6 + 0.8201 * x_7 + (-0.0178) * x_8 + (-0.6824) * x_9 + 0.6575 * x_{10} + (-0.0204) * x_{11} + (-0.4687) * x_{12} + 0.4181 * x_{13} + (-0.0271) * x_{14} + (-0.2207) * x_{15} + 0.1815 * x_{16} + (-0.0732) * x_{17} \quad (5.6)$$

The coefficients of Eq. (5.6) are obtained from the trained linear regression machine learning model. The  $x_n$  in Eq. (5.6) stands for the detected hourly SEU number from the monitor  $n$  hours ago. Thus, the above prediction function can start to predict after the monitor consecutively works and records the first 17 hours of data. Since this design is intended to be used as an embedded part of the space-borne system, simplicity and flexibility are essential concerns. Thus, to avoid floating-point calculation and reduce the hardware complexity, the coefficients in Eq. (5.6) are magnified by  $2^n$  times and only taking the integer part to simplify the equation. The magnification factor needs to ensure that the new prediction equation-induced accuracy variation is less than 1%. In this study, the magnification factor 32 is used, and the corresponding prediction function is as follows:

$$SEU_{pred\_acc\_32} = (41 * x_1 + 3 * x_2 + (-38) * x_3 + 33 * x_4 + 3 * x_5 + (-29) * x_6 + 26 * x_7 + (-1) * x_8 + (-22) * x_9 + 21 * x_{10} + (-1) * x_{11} + (-15) * x_{12} + 13 * x_{13} + (-1) * x_{14} + (-7) * x_{15} + 6 * x_{16} + (-2) * x_{17}) / 2^5 \quad (5.7)$$

Two 32\*21-bit address register files are used for logging the historical SEU data and prediction function coefficients. Regarding the historical SEU data register file, a single 21-bit entry consists of a valid entry bit and a 20-bit representing the number of detected upsets. According to historical solar events analysis for the solar cycle 24, which is introduced in Section 5.2, the peak value for the hourly upsets count of the target SRAM is 118122 upsets/hour/2Gbit. Therefore, the size of the selected register file can guarantee regular data storage even during large SPE peak fluxes. Moreover, up to 32 historical hourly upsets records can be thus stored simultaneously. The oldest individual record can be automatically discarded if the register file overflows. The contents of the coefficients register file are loaded during the system setup and updated after the online parameter adjustment procedure described in Section 5.4.2. After being magnified, as shown in Eq. (5.7), the coefficients are stored in each row separately.

The accumulator performs repeated addition calculations, thus implementing the multiplication operation in the prediction function. Therefore, a much longer calculation time than the traditional multiplier is expected. In this study, for Eq. (5.7), a total of 262 clock cycles is needed in the accumulator. Therefore, the minimum required time for Eq. (5.7) calculation is 5.24  $\mu$ s when the working frequency is 50 MHz. As the historical SEU data register file is updated on an hourly basis (i.e. the calculation is required only once every hour), the calculation speed for the



accumulator is sufficient for the analyzed application. The accumulator contains a 32-bit full adder, one two's complement number converter and a 32-bit register. The register keeps the intermediate arithmetic result from the adder. The inputs for the adder are the selected  $x_n$  and previous results from the register. Moreover, the selected  $x_n$  is converted to the two's complement form when the corresponding coefficient identifies a subtraction operation. Considering the calculation in practical applications, the overflow is not expected.

The control logic processes the coefficients in order to select the appropriate  $x_n$  for the accumulator, and to determine the number of repetitions in SEU prediction mode. The right shifter is used to shrink the calculation result based on the previous magnified factor, which is a 5-bit right shift for Eq. (5.7).

#### 5.4.2 Online Parameter Adjustment

The SGD algorithm is implemented to update the SEU prediction model parameters, thus, adapting to the changing working environment. Due to the simplicity and low-cost design, the implementation of the SGD algorithm reuses the same hardware design as the SEU prediction but with new control logic in the current operation mode. When activated, the online parameter adjustment is performed automatically after a new detected hourly SEU data is received.

According to the SGD algorithm, the update of each parameter  $Coef_j$  in the prediction function is based on the following equation:

$$Coef_j := Coef_j - \alpha \frac{\partial J(\theta)}{\partial Coef_j} \quad (5.8)$$

where  $\alpha$  is the learning rate and  $\frac{\partial J(\theta)}{\partial Coef_j}$  is the gradient of the cost function with respect to the model parameter  $Coef_j$ . For the SEU prediction Eq. (5.6), the corresponding learning rate is set to 0.02. Thus, in order to avoid processing floating-point data, magnification by  $2^n$  times for the learning rate is required. As an example, this thesis selected the magnification factor 256 for the learning rate. The partial derivative function  $\frac{\partial J(\theta)}{\partial Coef_j}$  could be defined as:

$$\frac{\partial J(\theta)}{\partial Coef_j} = \frac{\partial (h(x) - y)^2}{\partial Coef_j} = 2(h(x) - y)x_j \quad (5.9)$$

where the  $h(x)$  denotes the predicted SEU data from the selected prediction function,  $y$  is the detected corresponding SEU data and  $x_j$  stands for the detected hourly SEU data from the monitor in  $j$  hours ago. Due to the limitation of hardware resources, the online updating of one prediction function parameter  $Coef_j$  is broken down into the following steps:

- 1) Calculate the error between predicted and observed SEU values  $E$ ,  $E = h(x) - y$ ;
- 2) Obtain the corresponding gradient of the cost function  $G_j$ ,  $G_j = 2 * E * x_j$ ;
- 3) Multiply by the learning rate,  $L_j = \alpha * G_j$ ;
- 4) Update  $Coef_j$ ,  $Coef_j := Coef_j - L_j$ .

In order to update all parameters, i.e. for Eq. (5.7), the above steps need to be repeated 17 times. All required calculations are performed by repeated addition/subtraction in the accumulator and shift register. The overall calculation time greatly depends on the size of  $x_j$  in step 2). However, even if all  $x_j$  reaches the maximum number, 118122 upsets/hour/2Gbit, the required clock cycle to perform all the calculations is less than 3 M cycles, which corresponds to 0.06 s when the working frequency is 50 MHz. Although the online parameter adjustment requires a much longer time than the forecast of SEU, it is still appropriate for this study since the calculation is only needed to perform once per hour.

## 5.5 Analysis of Results

### 5.5.1 Prediction Performance Analysis

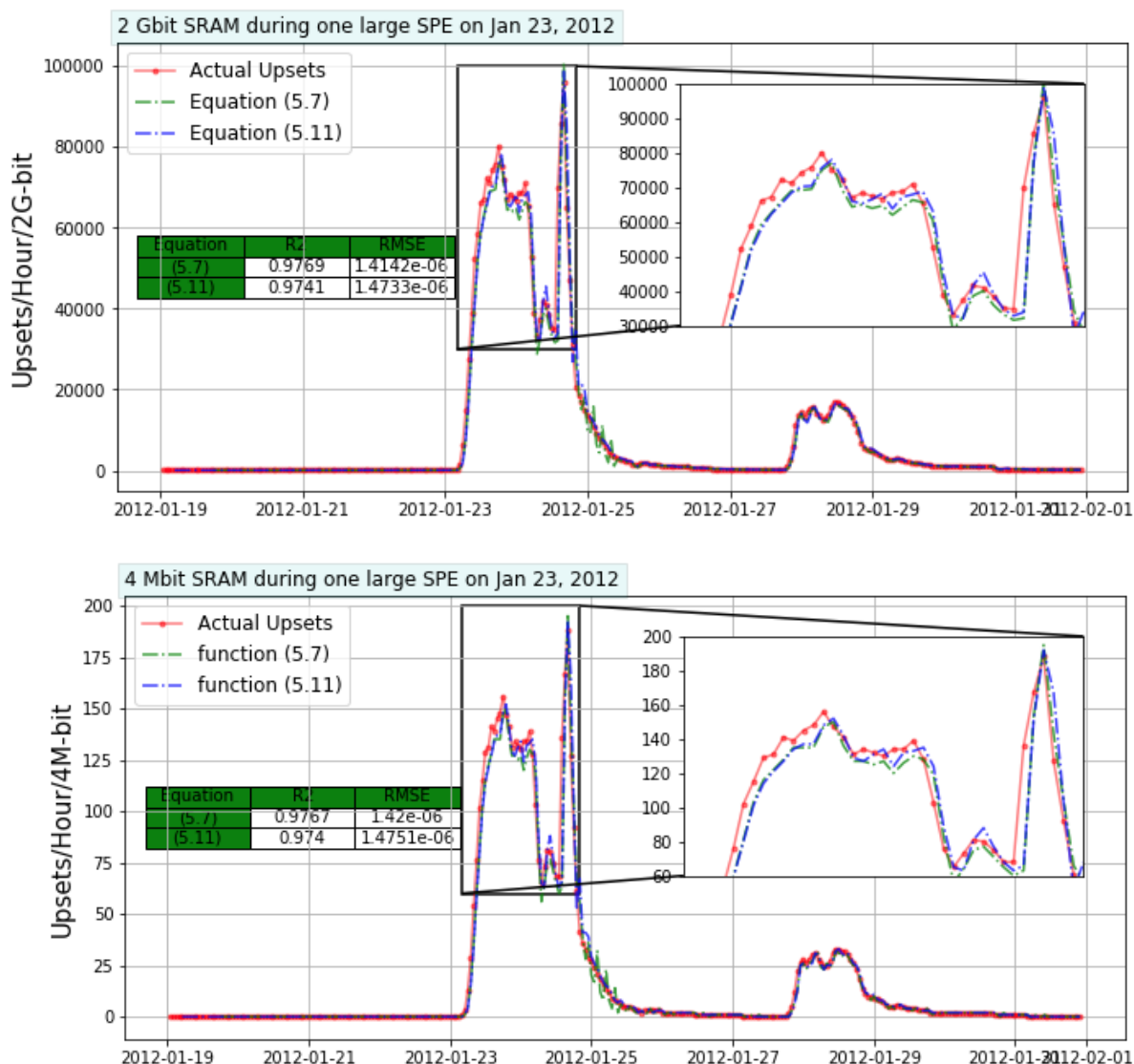
This section analyses the impact of SRAM size and history data size on the SEU prediction performance. The analysis in Section 5.3 is done for a large size SRAM with a size of 2 Gbit and with a history data length of 17. However, many embedded systems do not have multi-Gbit SRAM resources but relatively much smaller internal SRAM with the size from several Mbit to tens of Mbit. In such a case, the small detection area of the SRAM may not provide sufficient sensitivity, and it is necessary to evaluate the optimal SRAM size required for particle detection. Moreover, the selection of history data length of 17 means the prediction cannot be made for the first 17 hours, which may be too long for some scenarios where faster prediction is required. According to the history data length analysis in Section 5.3, length four also has an excellent  $R^2$  score with a slightly worse RMSE performance. For this case, the prediction equation is:

$$SEU_{pred\_fast} = 1.1939 * x_1 + 0.1105 * x_2 + (-0.7789) * x_3 + 0.4478 * x_4 \quad (5.10)$$

The magnification factor 1024 is used for the above function, thus, the corresponding function implemented in the hardware accelerator is:

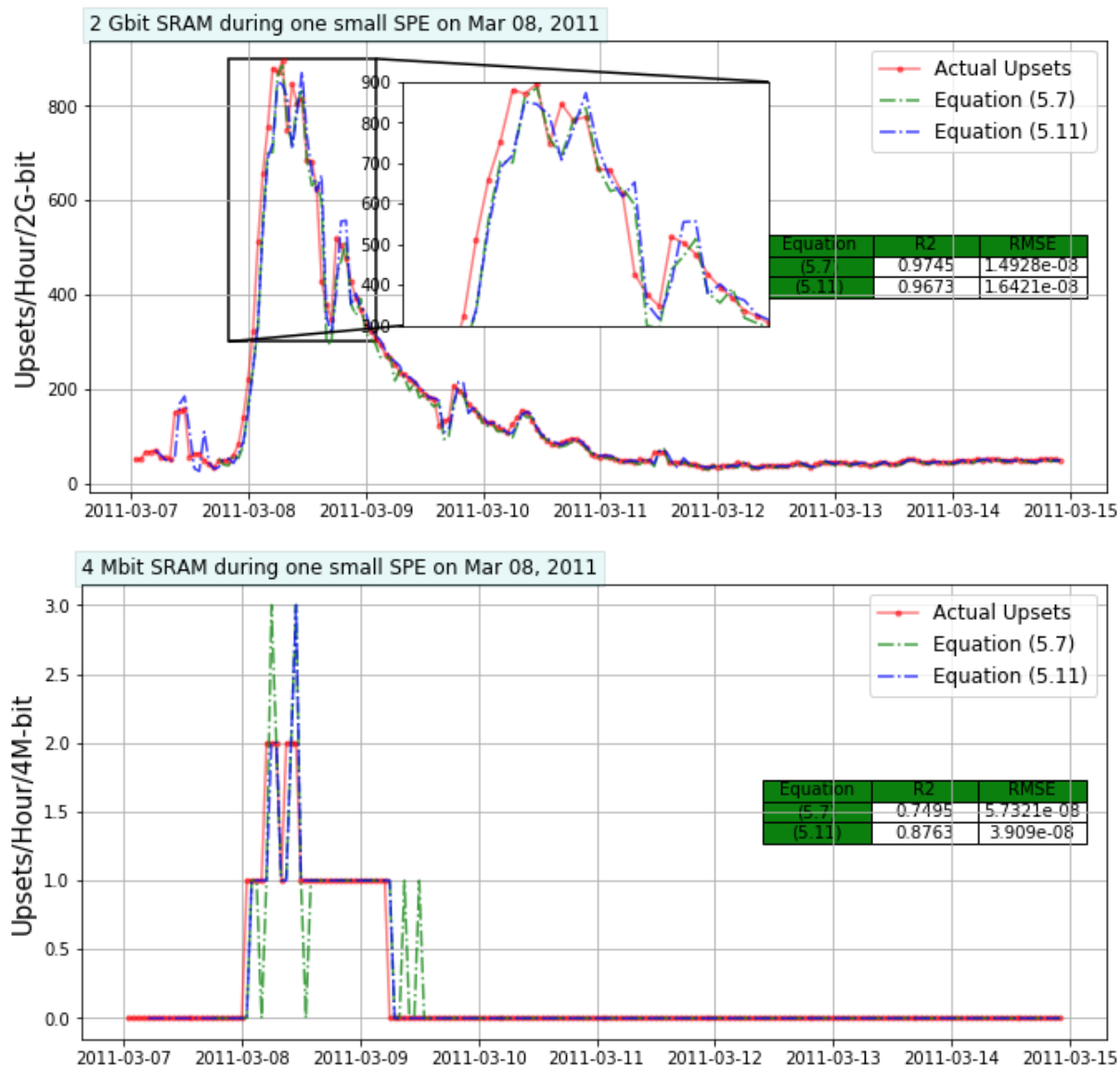
$$SEU_{pred\_fast\_1024} = (1223 * x_1 + 113 * x_2 + (-798) * x_3 + 459 * x_4) / 2^{10} \quad (5.11)$$

Figures 5.23 and 5.24 show the hardware accelerator SEU prediction performance based on Eq. (5.7) and (5.11), for 2 Gbit and 4 Mbit SRAMs, during large and small SPEs, are illustrated. The SEU rate for 4 Mbit SRAM was determined by scaling the SEU rate for 2 Gbit, i.e. the scaling factor is determined as the size ratio of the two SRAMs. This approach is adopted because the SEU rate is roughly proportional to the SRAM produced in the same technology and using the same bit-cell architectures [KRS<sup>+</sup>15]. However, it is essential to mention that this is a rough estimate because the cross-section may differ among different SRAMs. In order to facilitate the comparison of the evaluation scores, the RMSE scores in Figures 5.23 and 5.24 are scaled with the corresponding SRAM size.



**Figure 5.23:** Jan 19-31, 2012, hardware accelerator SEU prediction performance for 2Gbit and 4 Mbit SRAM during one large SPEs, respectively.

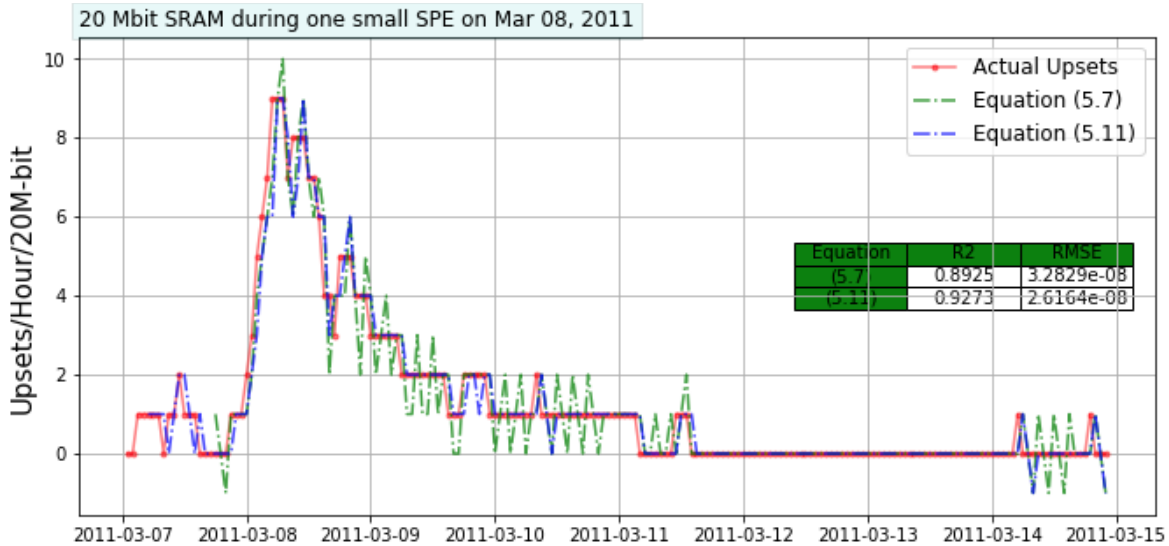
It can be seen that Eq. (5.7) and (5.11) can predict the SEU variation reasonably accurately for the small and large SPEs with 2 Gbit SRAM. However, in the case of 4 Mbit SRAM, only the large SPE can be observed. Neither of these equations can provide sufficient accuracy during small SPE. The main reason is that the SEU monitor with 4 Mbit SRAM does not have sufficient resolution to provide valid SEU data for prediction during the SPE onset period.



**Figure 5.24:** Mar 07-15, 2011, hardware accelerator SEU prediction performance for 2Gbit and 4 Mbit SRAM during one small SPEs, respectively.

In Figure 5.25, the prediction performance for 20 Mbit SRAM during the same small SPE as previous is shown. It can be observed that the 20 Mbit SRAM can predict small SPEs. In addition, since not much low-resolution historical SEU data is considered, Eq. (5.11) has a better prediction performance than Eq. (5.7). In order to get a smoother prediction curve than in Figure 5.25 and thus ensure good quality of SPE prediction, a larger SRAM needs to be used. Moreover, to generate the algorithm's reliability in any conditions, especially to the small sensitivity to small changes in space flux, the proposed SRAM-based monitor is supposed to detect at least one SEU each hour, even during the background space radiation condition. Therefore, according to the cross-section parameters of the target SRAM from the radiation test and the CRÈME 96 suit, the corresponding minimum SRAM size can be collected. For the target 65 nm SRAM, to guarantee the above

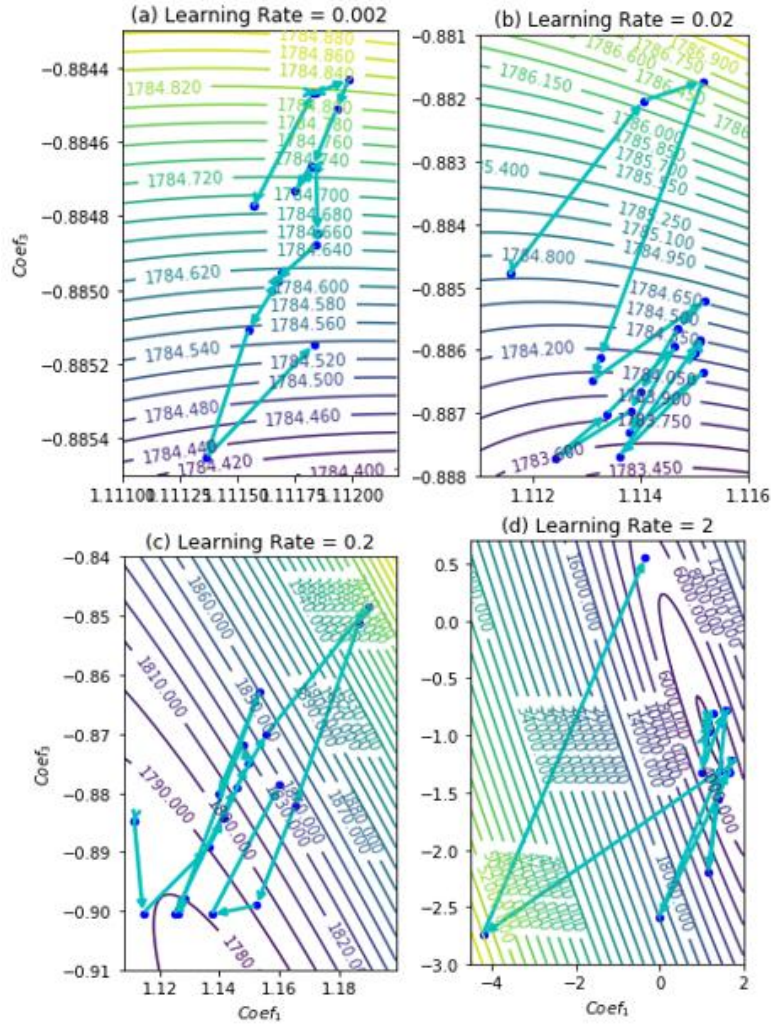
requirement, around 35 Mbit SRAM is recommended for hourly SEU detection. For certain cross-section data, the larger the SRAM size (i.e. the detection area), the more radiation particles can induce SEUs in each detection period. Thus, the more sensitive the variation in space radiation can be obtained.



**Figure 5.25:** Mar 07-15, 2011, hardware accelerator SEU prediction performance for 20 Mbit during one small SPEs.

### 5.5.2 Online Learning Performance Analysis

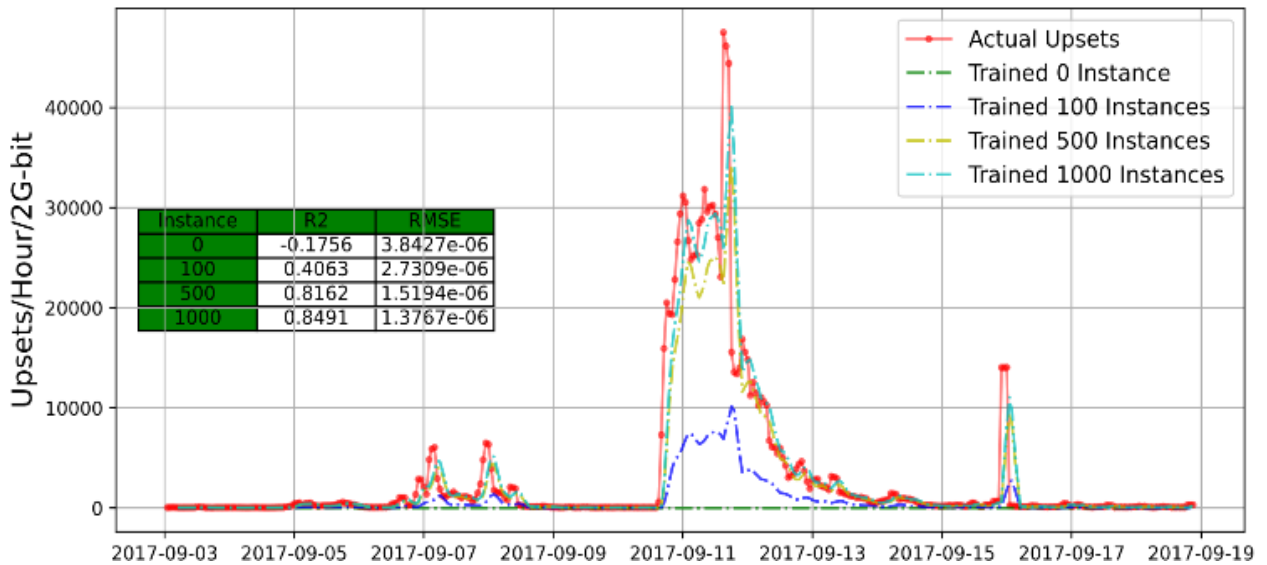
The learning rate is a critical parameter for the SGD algorithm for the online learning process, which determines how fast the system adapts to the changing data. The high learning rate allows the system to adapt to new inputs quickly. However, an excessively high learning rate may induce the system to forget the old data, thus, reducing accuracy quickly. Conversely, a low learning rate makes the system less sensitive to changes in new data. Thus, evaluating an appropriate learning rate for the online parameter adjustment is necessary. Moreover, applying online learning to obtain the SEU prediction function from scratch is also analyzed, which is essential when the offline training data is unavailable, such as in deep-space missions. In addition, since the online environment may be inconsistent with the offline analysis assumptions, the predictive optimization of online learning under different assumptions is also analyzed.



**Figure 5.26:** The gradient descent with contour plot with respect to the  $Coef_1$  and  $Coef_3$  under the different online learning rates.

The influence of the learning rate on the online parameter adjustment of the predictive prediction equation is analyzed. For the target prediction function, each item contains its corresponding parameter coefficient,  $Coef_n$ . The general idea of the SGD algorithm is to adjust these  $Coef_n$  with the selected learning rate on each instance. Figure 5.26 presents the gradient descent with the contour plot of the online learning cost function with respect to the SEU prediction coefficients  $Coef_1$  and  $Coef_3$ , which are the two most weighted coefficients in the prediction function in Eq. (5.6) and (5.10), regarding the different learning rates. The RMSE cost function is used to evaluate the prediction performance analysis. The numbers on the contour line stand for the expected RMSE prediction error with the corresponding coefficients. Thus, the smaller, the better for the model performance. The online learning process uses the same method as introduced in Section 5.3.4, in which 50% of the data set is used to get the initial linear regression offline training model and the remaining 50% data set is used for online learning with different learning

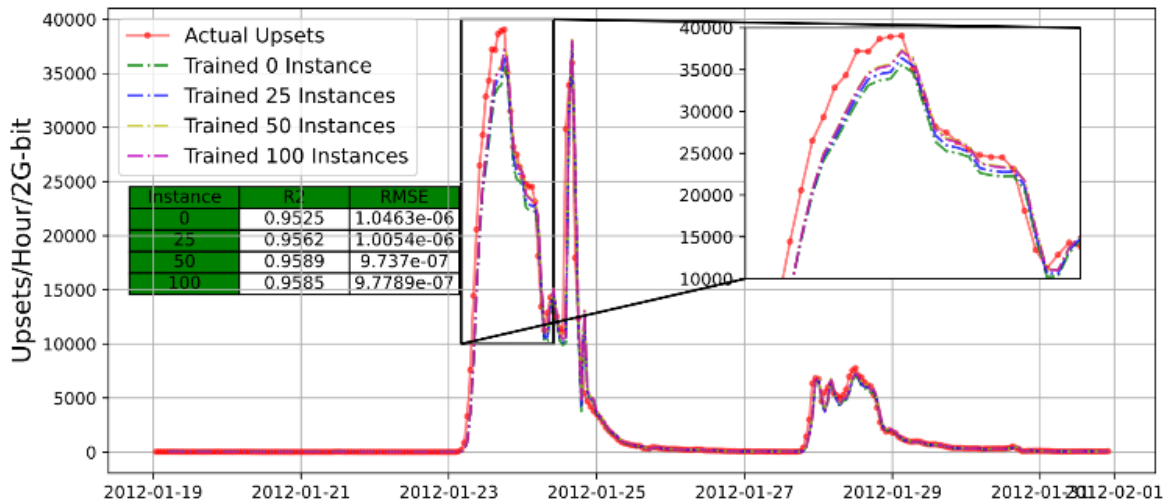
rates. Four online adjustments with different learning rates are performed in Figure 5.26, and all of them have the same starting point, where  $Coef_1$  is 1.11154 and  $Coef_3$  is -0.88478. As the online adjustment progresses, although there would be fluctuations, the prediction errors in Figures 5.26 (a) and (b) are steadily decreasing. However, as the learning rate increases in Figure 5.26 (c) and (d), the online adjustment could cause a large prediction error, which induces a cliff drop in the prediction performance. The main reason is that a large learning rate may cause abnormal data to impact prediction performance excessively. In addition, online learning is based on the results from offline training, thus, a small learning rate is more suitable for this study. However, a too-small learning rate could cause a much longer time to adapt to the new working environment. Thereby, the learning rate of 0.02 is selected in this thesis. Moreover, the adaptive learning rate method [YCC95] can also be used in the online training progress, reducing the learning rate based on a pre-defined schedule method, such as time-based decay, step decay, etc.



**Figure 5.27:** Online learning prediction on a middle-size solar event performs from scratch (i.e. no offline dataset available). The initial coefficients of the prediction equation (i.e. trained 0 instances) are set to all 0.

Since the SGD algorithm supports learning from the new data on the fly, the training approach to abandon offline learning and directly apply online learning to form a prediction function from scratch is possible. It is essential for the scenarios where offline data is missing, such as deep-space missions. In Figure 5.27, the prediction performance with the online learning on several amounts of instances for 2 Gbit SRAM is illustrated. The training starts from a linear regression model in which all the coefficients are set to 0. With constant inputs of detected SEU data, the SGD algorithm is used to update coefficients after collecting each data instance. In this example, it can be seen that after the SGD algorithm takes about 1000 instances, a relatively well-performing

SEU prediction equation can be obtained. Due to the requirement for quick adaptation to a new environment, a high learning rate is expected in the initial stage. However, as the prediction error continues to decrease, a gradual reduction of the learning rate should perform. Moreover, since the SPE phenomena do not frequently occur, a purely online learning process from scratch could take a long time to achieve good prediction accuracy for space applications.



**Figure 5.28:** The prediction performance for online learning optimizes the offline prediction function on the new data set (i.e. with different shielding performance).

The offline training is based on a limited data set and certain parameter assumptions, which may not be the same as the online condition. Thus, in the actual application, many other possible reasons would affect the accuracy of the offline algorithm, such as actual shielding parameters, changing of the satellite orbits, variation of solar cycles, etc. Due to the time-consuming acquisition and processing of spatial ion flux data, thus, it is not possible to comprehensively display the performance of online learning under all various possible changes in the thesis. As an example, the shielding parameter influence of online learning is analyzed in this thesis. During the offline data set collection phase in Section 5.2.3, 100 mils of aluminium shielding is assumed. However, the actual shielding parameters during the online prediction phase may not be the same. In order to analyze the shielding impact, a new data set has been collected as the same approach in Section 5.2, but with a new aluminium shielding parameter, 10 mm (394 mils), which is used in [HED04]. The prediction function Eq. (5.7) and the online learning algorithm evaluate the new data set. Figure 5.28 shows the new data validation results, and the trained 0 instance means the prediction performance when only using Eq. (5.7) for the new data set evaluation. The results show that the prediction based on Eq. (5.7) declines during the SPE peak period, and with online learning assistance, the forecast results can be gradually improved. Detailed analysis of online learning



performance in various environments will be the future work, such as with different orbits, solar cycles, offline analysis parameters, etc.

### 5.5.3 Synthesis Results

As the overall idea is to implement the SRAM monitor and hardware accelerator together with the target system on a single chip, it is necessary to investigate the introduced power and area overhead. The synthesis results presented in this section have been obtained for the IHP's 130 nm bulk CMOS technology with a supply voltage of 1.2V and an operating frequency of 50 MHz. Although the synthesis analysis in this section uses different technology than the analyzed SRAM, the results are of significant value for hardware consumption comparison because the proposed design is general and can be implemented in different technologies. The choice of the target technology will define the SRAM's cross-section, which is obtained from irradiation experiments.

**Table 5.4:** Area and power comparison of the proposed space environment hardware accelerator with a 20 Mbit SRAM and the proposed SEU monitor.

	Area (in $mm^2$ )	Power (in $mW$ )
<b>20 Mbit SRAM</b>	14	384
<b>SEU Monitor</b>	0.0957	0.211
<b>Hardware Accelerator</b>	0.6363	3.629

In Table 5.4, the total area and power consumption for 20 Mbit SRAM, SEU monitor and proposed hardware accelerator are shown. Although the power consumption of the proposed hardware accelerator design is about 18 times larger than that of the SEU monitor, compared with 20 Mbit SRAM, the induced area and power consumption are only 4.55% and 0.95%, respectively. Moreover, regarding the hardware accelerator, two 32\*21-bit address register files are one of the main contributors to the area consumption, which can be reduced in the real case. Besides that, the power consumption for the proposed design is the pessimistic estimation, which is based on the assumption that some percentage of flip flops is switching in each clock cycle. However, in the proposed hardware accelerator, this calculation can take some number of cycles per hour, and after that, the module could be clock gated. Therefore, the energy consumption of this module would be orders of magnitude lower. Thus, these results indicate that the cost and overhead for the hardware accelerator are negligible compared to the host SRAM.

## 5.6 Summary

This chapter proposes an approach for the in-flight SPE and SEU prediction of the SRAM-based SEU monitor system in space-borne electronic systems, which can forecast the space environment and is suitable for a highly dependable self-adaptive resilient system employed in space applications. An SRAM-based SEU monitor is used to analyze the space environment

autonomously. The proposed solar condition predictor method can collect the in-flight real-time SEU rate data and a pre-defined prediction model from forecasting the following SEU rate, thus, determining the corresponding solar conditions. Thus, the upcoming flux variation, and the corresponding SPE, can be predicted from the rise of the SEU count rate at least one hour in advance. Moreover, the fine-grained hourly tracking of the SEU variations during the SPE and under normal conditions is supported.

In order to perform a comprehensive analysis of the historical solar events, a total of 36 SPEs, which occurred in solar cycle 24 (2008-2019), are selected and analyzed in detail. As a result, over 5000 hourly SEU rate data of a 65 nm SRAM are collected and used to train several machine learning models. In this thesis, five existing machine learning regression models with the time-series method have been trained and evaluated to select the optimal model on the collected historical solar event data set. With the comparison of the prediction accuracy and the implementation complexity of the embedded design, the linear regression model is selected to be used as the prediction model. The selected linear regression can provide fine-grained predictions of hourly SEU rate with high accuracy at least 1 hour in advance. Moreover, an online learning method has been introduced, which can further improve the prediction accuracy of machine learning in real-time. Online adjustment allows adapting the system to completely unexpected situations that could not be predicted based on offline training.

A dedicated hardware accelerator is customized to execute the proposed machine learning prediction model and online learning algorithm. Analysis has shown that the proposed hardware accelerator has an outstanding prediction accuracy for the analyzed application and with a negligible cost. The proposed space environment prediction method is intended for a highly dependable and self-adaptive resilient system employed in space applications, which can trigger the optimal SEU mitigation methods under variable radiation conditions, as introduced in Chapter 6. Therefore, the target system can drive the appropriate radiation hardening mechanisms before the onset of high radiation levels.

## **Chapter 6**

# **Self-Adaptive Fault Resilience System**

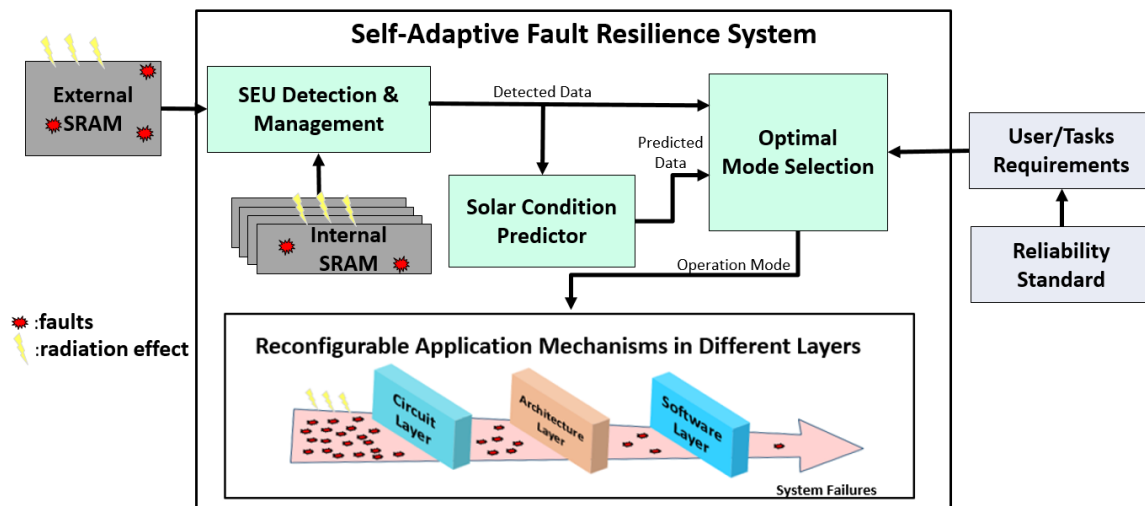
Radiation-induced fault mitigation is a mandatory step for reliability-critical space-born applications. However, common static fault mitigation methods are generally based on the worst-case assumptions, which are most of the time unnecessary and result in resource overhead. Moreover, the system reliability, performance, and power consumption requirements could vary depending on tasks and working environments. To overcome this problem, a Self-Adaptive Fault Resilience (SAFR) system is introduced in this chapter, which can collaboratively and efficiently mitigate radiation-induced faults on a highly dependable multiprocessing system and achieve the trade-off between reliability, power consumption, and performance in real-time. Moreover, multiprocessing systems have inherent hardware redundancy and are convenient for deploying reconfiguration mechanisms. Therefore, various multiprocessing operation modes can be formed, such as fault-tolerant, low-overhead, etc. The proposed system can use the detected and predicted radiation environment information to select the optimal operating mode flexibly in real-time.

This chapter is divided into five sections. Section 6.1 introduces the concept of the SAFR system. Section 6.2 presents a few existing resilience methodologies on different system layers. The method of optimal mode selection is detailed in Section 6.3. The evaluation and discussion are presented in Section 6.4. The chapter is concluded in Section 6.5.

### **6.1 Concept**

The concept of the proposed SAFR system is depicted in Figure 6.1. This platform comprises the SEU monitor network, the solar condition prediction module, a range of reconfigurable resilience mechanisms in different system layers and an optimal mode selection module. The SEU monitor network, detailed in Chapter 4, gathers the in-flight SRAM SEU rate data, thus, and detects the real-time radiation environment. The solar condition prediction module, detailed in Chapter 5, collects the detected SEU data to forecast solar conditions and predict the SRAM SEU rate. In embedded applications, multiprocessor architectures started to be the dominant trend, and it is convenient and common for applying various reconfigurable mechanisms on different system

layers [SVK<sup>+</sup>16] [JCK18]. For example, dynamic methods such as Dynamic Voltage Frequency Scaling (DFVS), clock- and power- gating can be used at the circuit layer, the core-level N-Module Redundancy (NMR) can be deployed at the architecture layer, and dynamic task scheduling can be deployed at the software layer, etc. Therefore, various multiprocessing operation modes can be formed, such as fault-tolerant modes (e.g., core-level triple module redundancy) or low-overhead modes (e.g. high performance and low power consumption). Different operation modes may affect system reliability, power consumption and performance. In this paper, the main focus is on system reliability while also considering system power consumption and performance. The optimal mode selection module is composed of a series of pre-defined operating mode triggering thresholds, which determine the appropriate output operation mode from the detected and predicted space environment as well as the user/task requirements.

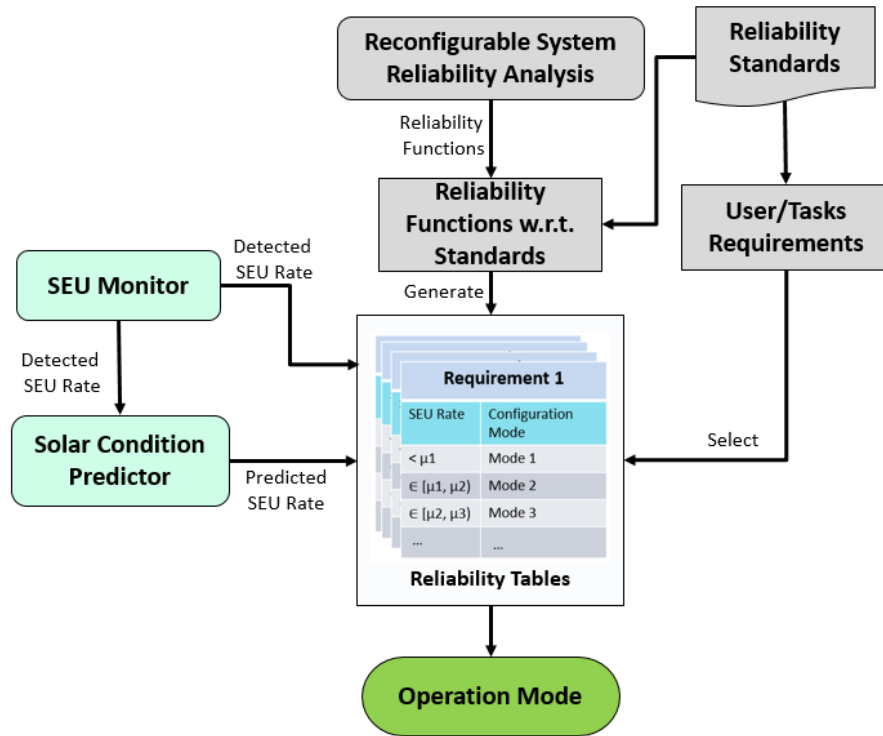


**Figure 6.1:** General concept of Self-Adaptive Fault Resilience system. The green blocks represent the main objectives of this dissertation.

Fault resilience is the property that keeps the system not deviating from the correct operation in the presence of faults. There are a lot of existing fault resilience mechanisms improving system reliability, and most of them focus on the improvement of a single abstraction layer in the system design. Instead of improving the efficiency of the individual layer's resilience mechanisms, the more promising approach to achieving an effective and reliable system operation is to utilize the different available approaches or parameters at multiple abstraction layers and combine them to optimize the design in a cross-layer manner. Moreover, by utilizing different available dynamic mechanisms, the cross-layer methodology is also a promising way to achieve the trade-off between system reliability, performance, and power consumption. The cross-layer resilience system has the potential to achieve higher average performance, more reliable operation, lower cost, and energy

consumption by taking advantage of the information and resources available methods across different system layers.

In this thesis, the dynamic use of multiprocessing systems is in focus. Such systems are the backbone of the modern embedded system. The SAFR system applies the inherent advantage of hardware redundancy of multiprocessor systems and introduces two common-used existing reconfiguration mechanisms in this thesis (i.e. core-level NMR and DVFS) as the example, detailed in Section 6.2. Since the working environment and task reliability requirements of the system may be different under different conditions, these two mechanisms can form a variety of system operation modes with different system reliability, power consumption and performance. For example, a high-reliability mode can be processed during harsh radiation conditions; a high-performance mode is suitable during image transmission tasks in which a few faults could be accepted, etc. Therefore, selecting the optimal set of these available dynamic mechanisms is essential for the best trade-off between system reliability, power consumption, and performance under different working conditions and task requirements.



**Figure 6.2:** Decision tree for determining the operation mode. The grey blocks are the main focus of this chapter.

The decision tree for determining the optimal operation mode for the SAFR system is shown in Figure 6.2. The static reliability analysis for the target system with various operation modes should be performed during the system's design phase. Therefore, corresponding reliability functions in

different modes can be collected, and the relationship between the SEU rate and the system reliability variations can be displayed, as detailed in Section 6.3. Moreover, the connection between these functions and the predetermined reliability standard can be established. In this thesis, as an example, the safety standard based on the IEC 61508 standard is used [IEC05]. Four Safety Integrity Levels (SIL) are defined in the target standard, with SIL 4 being the most dependable and SIL 1 the least. The SIL is defined as a relative risk reduction level supplied by a safety function or as a goal level of risk reduction. Therefore, for the IEC 61508 standard, four reliability tables can be formed from the static analysis to represent the connection between the SEU rate and operation modes under the different reliability requirements from each SIL level. In order to satisfy the SIL demand, the system can launch a specific operation mode within a certain SEU rate range. Based on the real-time detected and predicted SEU rate information from the proposed monitor and forecasting system, as well as the required system reliability, the operating mode can be determined and launched by checking the collected reliability tables. There may be many different user/task requirements, so it is hard to demonstrate them comprehensively. Thus, an overall requirement is assumed, and the corresponding reliability table calculation procedure is performed in Section 6.3. Moreover, using the space radiation flux data during solar cycle 24, the performance of the proposed SAFR system and the corresponding optimal mode selection strategy in historical conditions are analyzed in Section 6.4.

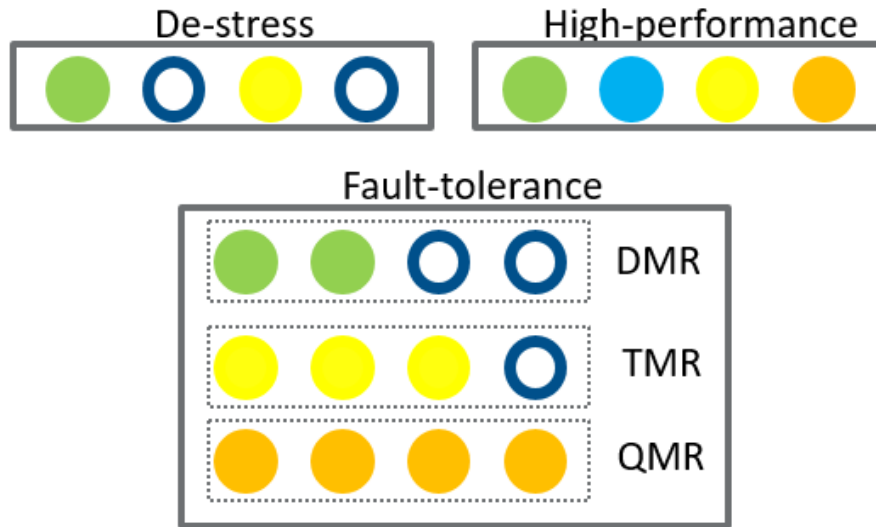
## **6.2 Reconfigurable Resilience Methods**

As introduced in Section 6.1, various existing dynamic resilience methods exist on different system layers. In this section, two of these methods, i.e. Core-level NMR and DVFS, which can be integrated into multiprocessing systems, are introduced in detail, respectively. Moreover, the corresponding system reliability, power consumption, and performance changes with different configuration modes are also analyzed.

### **6.2.1 Core-level NMR**

Core-level NMR is a reconfiguration method for multiprocessing systems, which enables the tightly-synchronized core-level formation of NMR groups of multiprocessor cores with voting in each clock cycle [ASI14]. The key concept of this technique is the dynamic adaptation of fault tolerance and performance to application needs, which can be accomplished at the lowest possible rate of aging and power consumption. The main component to achieve the core-level NMR design is a binary matrix-based programmable NMR voter for multiprocessors [SHK<sup>+</sup>12], where each processing core could be selected whether it takes part in voting. Moreover, this voter can provide the status of inputs and can change from 2MR to NMR systems with any combination of active processors, which is an NMR on-demand system. Besides that, the power- or clock-gating is also structurally embedded in the core-level NMR approach, which can be used to deactivate the cores.

The burden might be taken up dynamically by processing cores. As a result, the corresponding power can be conserved, and the multiprocessor's lifetime can also be extended [ASI14].



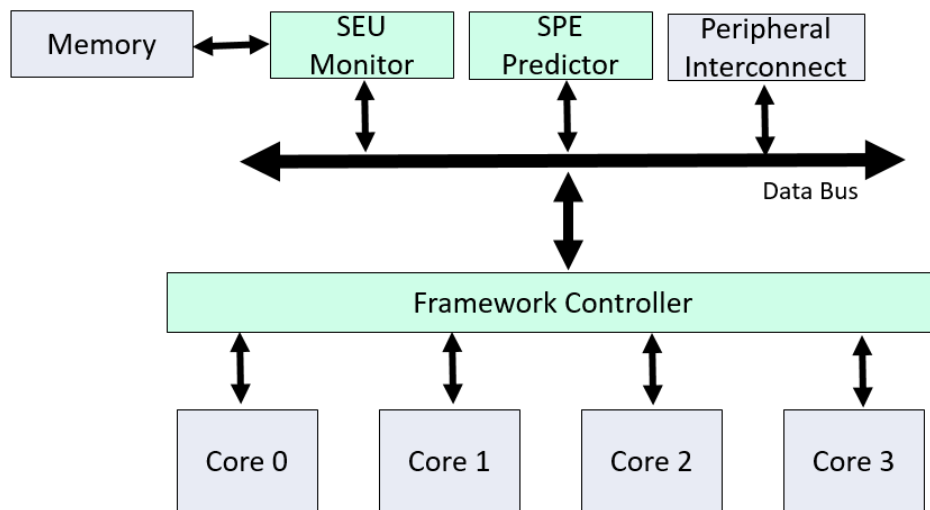
**Figure 6.3:** Example of possible core-level NMR operating mode in a quad-core multiprocessing system. The circle corresponds to the core. Four circles in the solid or dashed rectangle correspond to the quad-core system. The hollow circle represents the deactivated core with power-saving mechanisms. Different colours represent different tasks running on the cores.

By using core-level NMR design within the dynamically adaptable framework for the multiprocessing system [SSB+20], the target chip can provide four primary operating mode groups: de-stress, fault-tolerance, high-performance and mixed modes, as shown in Figure 6.3:

- De-stress mode: one or more cores operate with different tasks, while the others are clock-gated or powered-off. The purpose of the de-stress mode is to extend the multiprocessor lifetime and reduce power dissipation. Logically, less work reduces power consumption and aging (wear-out). Youngest-First Round-Robin (YFRR) core gating [SKK14] is an anti-aging strategy used to de-stress the operational core by moving the burden to a resting core. It is done on a regular basis and with the assistance of unique aging monitors integrated with each core [SKK11]. The YFRR method maps and schedules task(s) to the youngest core(s) in the multiprocessing system while turning off the oldest core. Moreover, the procedure of workload transfer from one core to another is similar to the general context-switch process.
- Fault-tolerant mode: it can enable fault tolerance for timing-critical tasks by providing hardware overhead in the system at the core level. The platform produces core-level NMR groups of tightly synchronized cores (lock-step) that execute each instruction concurrently, while NMR voting occurs in each clock cycle. The major building block of this process is a binary-matrix-based programmable core-level NMR voter [SHK<sup>+</sup>12]. By setting the

voter, the user can define which of the modules in the framework group constitute the NMR group since the module outputs are connected to the data inputs of the programmable voter. For example, in a framework group with four cores, the platform dynamically forms DMR, TMR, and QMR groups with any combination of cores, as shown in Figure 6.3. Moreover, this core-level NMR method can dynamically regroup the modules, such as upgrading from DMR to TMR, changing the combination of modules that form a TMR, etc.

- High-performance mode: the quad-core system operation corresponds to a common multiprocessor in this mode, i.e. each core does its own work. This mode can be chosen based on the needs of the application.
- Mixed mode: the three modes presented above are the basic operating modes. However, the primary modes can be mixed in any suitable way. For example, in a quad-core system, two cores can operate in DMR mode while the other core can run additional two different tasks simultaneously. It is a "natural" property of the architecture and does not require extra implementation overhead. The mode mixing is very convenient for systems with mixed criticality.



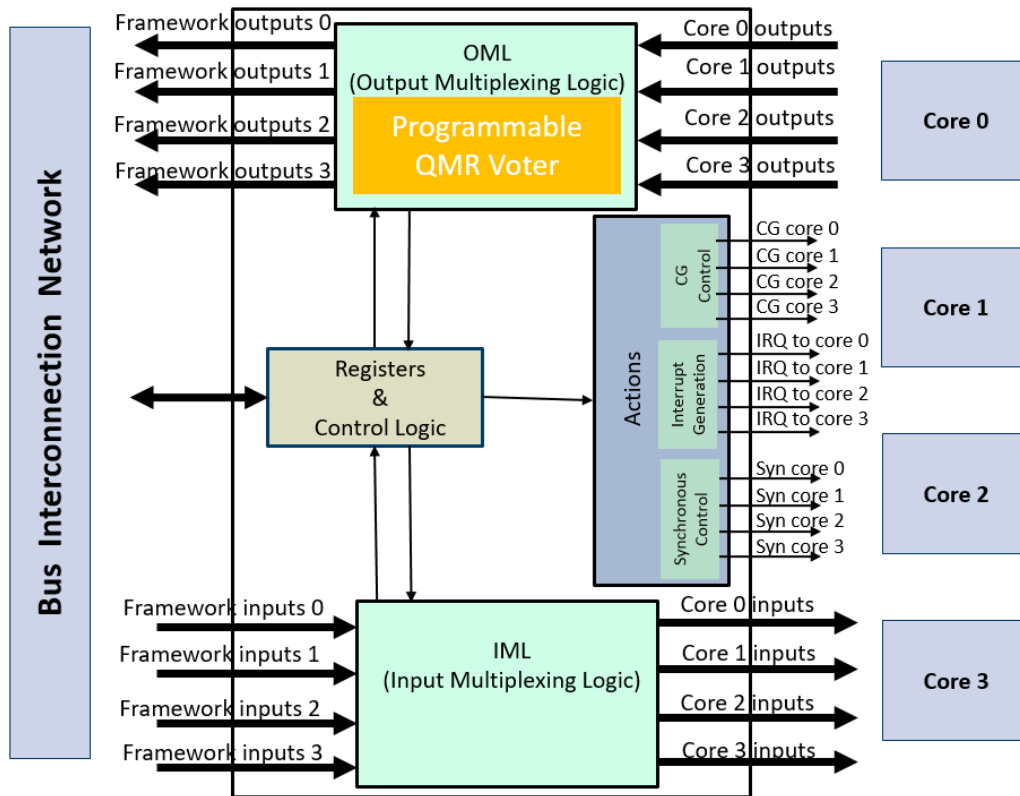
**Figure 6.4:** General architecture of a SAFR quad-core processing system. The core-level NMR modes are implemented and controlled in the framework controller. The green blocks represent the main focus of this dissertation.

## Implementation

In this thesis, in order to illustrate the operation of the SAFR on a multiprocessing system with the core-level NMR methodology, as an example, the above framework controller in one quad-core platform has been implemented, as shown in Figure 6.4. One binary matrix-based programmable QMR voter has been integrated as the core of the framework controller. Several control registers in the framework controller are used to define the configuration mode of the voter, as well as the



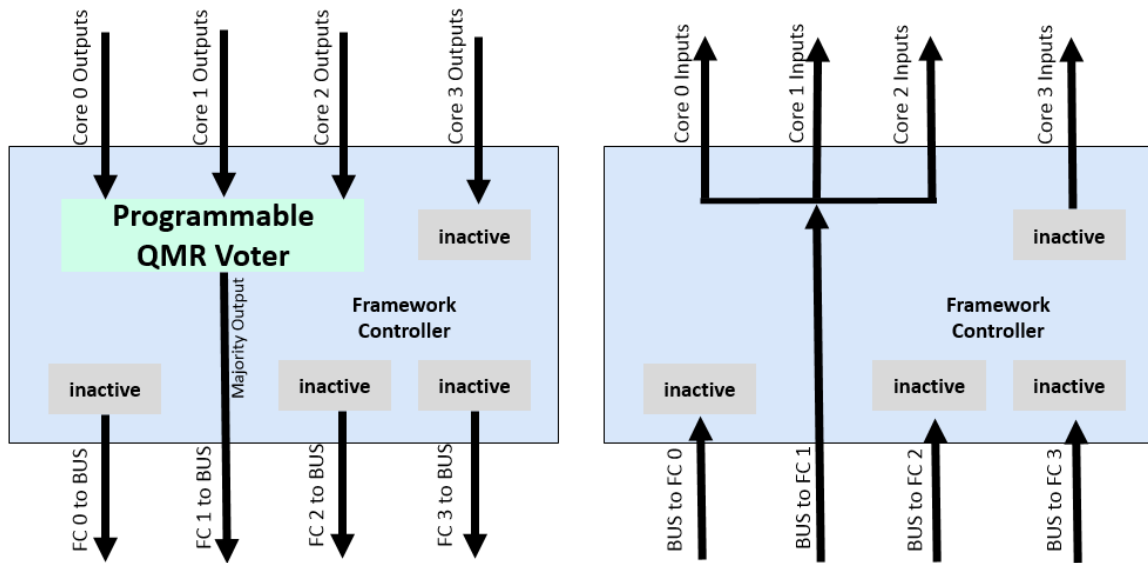
operating mode. The main data bus uses the interconnect tree structure, as shown in Figure 4.10, to access the monitor, memory banks, I/O, etc. The peripheral interconnect contains an APB bridge, which is a part of the Advanced Microcontroller Bus Architecture (AMBA) protocol family. Thus, it is very convenient to add a new peripheral as an APB slave for connection. The SEU monitor is embedded as the wrapper of memory banks, as introduced in Section 4.3.



**Figure 6.5:** General framework controller block diagram. Thick and thin lines denote data and control flow, respectively. CG stands for clock gating.

The framework controller block diagram is shown in Figure 6.5. The framework controller routes the inputs and outputs of all cores in the framework group, i.e. Output Multiplexing Logic (OML) and Input Multiplexing Logic (IML). The core outputs data are collected and compared by the programmable QMR voter in the OML when the fault-tolerant mode is selected. The framework inputs from the interconnection bus are managed by the IML and routed to the corresponding cores. There are several registers in the framework controller to determine the cores' QMR voter behaviour and settings. These registers are implemented as the slave for the APB bus controller. Thus, users can efficiently perform write or read these registers and control the framework's behaviour. One of the most important registers is the mode register, which defines the operating mode and the NMR groups by driving the programming bits of the core output programmable QMR voter [SHK<sup>+</sup>12]. For example, as shown in Figure 6.6, the mode

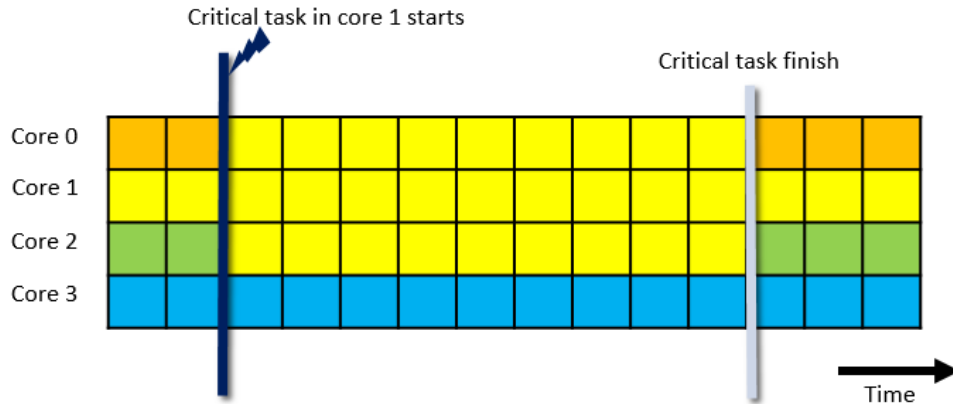
configuration of the framework under the above mode setting of TMR (0, 1, 2) with master core 1 is shown. The framework inputs and outputs of 0 and 2 are set to inactive level, the core 0, 1, and 2 outputs in the OML are sent to the QMR voter, and the majority output would be sent out. In the meantime, input 1 will be distributed to cores 0, 1 and 2, and core 3 is not affected by this operation. The outside can see a two-core system consisting of core 3 and a TMR group of cores. In addition, the above tight-synchronization of NMR groups is that state recovery (restoration) is easy to achieve by using the context-switch method.



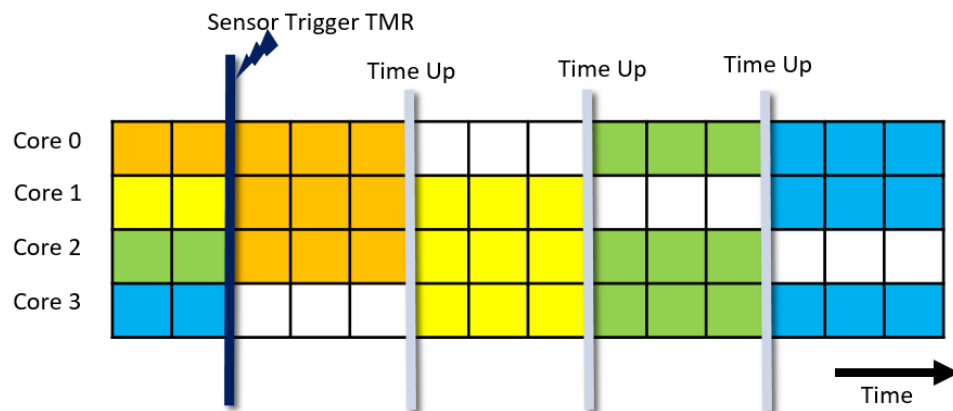
**Figure 6.6:** Example of OML, IML configuration in the framework controller. Core 0,1 and 2 form the core-level TMR, and core 3 is in the high-performance mode.

With the assistance of the above framework controller and the SEU detection/prediction system, two types of operation mode reconfiguration strategies can be applied: user-defined NMR and self-NMR. User-defined NMR aims to provide fault-tolerance protection for critical software, such as the attitude control process, even during background space radiation conditions. As an example, a user-defined TMR is shown in Figure 6.7, in which core 1 is dedicated to running critical applications. Thus, a TMR (0, 1, 2) can be formed for increasing reliability, and core 3 can still run its own less critical task. Various user-defined modes can be triggered by writing into framework controller registers from the processor. Unlike the situation discussed earlier, the self-NMR is intended to harden all the tasks during the harsh radiation condition, i.e. SPE, and can be automatically triggered by the SEU network. In this thesis, the focus is on the optimal mode selection for self-NMR. In Figure 6.8, a self-TMR mode in the quad-core system is shown as an example. After the SAFR system detects or predicts a harsh environment, the system automatically

triggers the appropriate self-NMR mode. Thus, all the tasks would be run with fault-tolerance protection within a specific time period.



**Figure 6.7:** Example of user-defined TMR in a quad-core multiprocessing system. Different colours represent different tasks running on the cores.



**Figure 6.8:** Example of self-TMR in a quad-core multiprocessing system. Different colours represent different tasks running on the cores. The hollow square represents the clock- or power-gating of the core.

### Reliability analysis

As shown in Figure 6.2, in order to perform the optimal mode selection, it is necessary to analyze the system reliability performance for different modes. Section 2.3.3 describes the system reliability evaluation approach. Moreover, the Eq. (2.2) also introduces the relevant system reliability as a function of time  $R(t)$ .

In a typical NMR system, the majority voter is the M-out-of-N decision-maker, where N voter inputs and at least M modules must be equivalent to create an error-free situation. Moreover, if  $M \leq N/2$ , an ambiguous situation could occur. For example, in the above quad-core system, if ( $O_0 =$

$O_1) \neq (O_2 = O_3)$ , where  $O_n$  stands for the core output signals. It is quite straightforward to assess the reliability  $R(t)$  of an NMR system. In probability theory, the binomial theorem might be used to derive as follow:

$$P (M \text{ success in } N \text{ trials}) = \binom{N}{M} p^M (1 - p)^{N-M} \quad (6.3)$$

where  $p$  is the likelihood of success, and in the case of a core-level NMR system,  $p$  denotes the system reliability function of a single core. Let each core have the same reliability  $R_c(t)$  and an exponentially distributed time-to-failure (i.e., Eq. 2.2). Thus,  $p = R_c(t) = e^{-\lambda_c t}$ , where  $\lambda_c$  stands for the target core failure rate. Moreover, considering that the reliability of the majority of the voter is  $R_v(t) = e^{-\lambda_v t}$ . Using Eq. (6.3), and assuming an exponential distribution of the time-to-failure as well as incorporating voter reliability, the reliability function of the generic NMR system with M-out-of-N voters can be obtained as:

$$R_{nmr}(t) = e^{-\lambda_v t} \sum_{i=M}^N \binom{N}{i} e^{-i\lambda_c t} (1 - e^{-i\lambda_c t})^{N-i} \quad (6.4)$$

Therefore, for the introduced quad-core core-level NMR system example, the following reliability functions with perfect majority voter can be obtained:

$$R_{high-performance}(t) = R_{de-stress}(t) = e^{-\lambda_c t} \quad (6.5)$$

$$R_{DMR}(t) = e^{-2\lambda_c t} \quad (6.6)$$

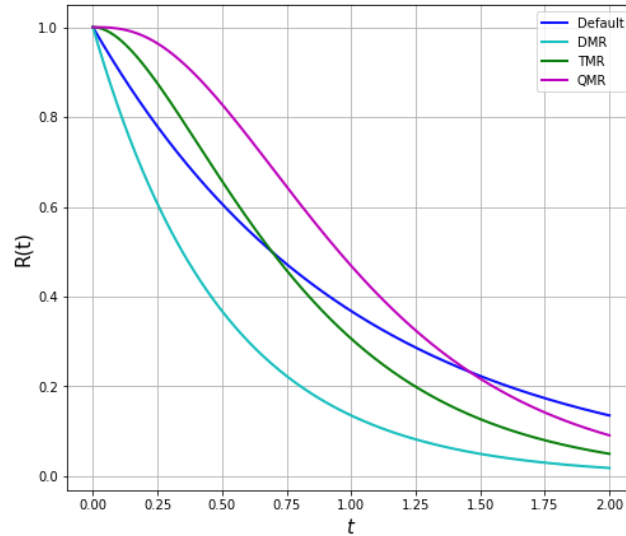
$$R_{TMR}(t) = 3e^{-2\lambda_c t} - 2e^{-3\lambda_c t} \quad (6.7)$$

It is essential to mention that each core runs a different program in high-reliability and de-stress modes. Therefore, the reliability of each core to perform the correct output is the same in both of these two modes. However, the general NMR system function (i.e. Eq. (6.4)) is not suitable for the core-level NMR system when the  $N > 3$ . The main reason is that the core could have plenty of output signals, and the error outputs of each core may not be the same. For example, in the target quad-core system, if  $O_0 = O_1 \neq O_2 \neq O_3$ , where  $O_n$  stands for the core output signals, the majority voter can still determine the correct output signals. Therefore, for the core-level NMR system ( $N \geq 4$ ), in order to obtain the correct output, the number of correct cores should be greater than the erroneous cores providing the same but wrong output signals. Therefore, for the QMR system, the corresponding system reliability function is:

$$R_{QMR}(t) = P_d 6e^{-2\lambda_c t} (1 - 2e^{-\lambda_c t} + e^{-2\lambda_c t}) + 4e^{-3\lambda_c t} - 3e^{-4\lambda_c t} \quad (6.8)$$

where  $P_d$  denotes the probability that one error core output is different from others error core outputs. The value of  $P_d$  can be obtained from the core-level fault-injection simulation experiences. Moreover, when  $P_d = 0.5$ , we can set the equation  $R_{QMR}(t) = R_{TMR}(t)$ . Otherwise, if  $P_d > 0.5$ , the QMR system has higher reliability than the TMR system, and in the opposite case, the TMR system is more reliable. Moreover, assuming a core has N output pins and core faults have the

same probability of generating the error signals on each pin, the  $P_d$  can be obtained as  $(1 - \frac{1}{2^{N-1}})$ . Figure 6.9 shows the  $R_{nmr}(t)$  equation in the target quad-core multiprocessing system from Eq. (6.5) – (6.8) for a module failure rate of  $\lambda_c = 1$ , which is one failure per  $10^9$  hours of operation, and the QMR is in the idealcase, where  $P_d$  is assumed to 1.



**Figure 6.9:** Comparison of reliability functions of core-level NMR in a quad-core system. The default means the de-stress or high-performance mode. The core failure rate  $\lambda_c = 1$  is assumed, and the QMR mode is shown as an ideal case ( $P_d = 1$ ).

The system performance parameter,  $Pe$ , is proportional to the number of parallel running tasks, and for example  $Pe_{high-performance} = 4Pe_{QMR}$ . In order to compare different modes with respect to power consumption,  $P$ , we need to consider both are two types of power consumption sources: dynamic and static components. The charging and discharging of the load capacitance of each gate causes dynamic power, while the leakage current mostly generates static power. Furthermore, while the core is idle, leakage current runs through the circuit and uses leakage power equal to around 3% of its active state [SJP<sup>+</sup>08]. In this thesis, for simplicity, the power consumption of cores during the sleep state is neglected. Therefore, the power consumption of each mode is proportional to the number of active cores, such as  $Pe_{TMR} = 3Pe_{de-stress}$ , etc.

## 6.2.2 DVFS

DVFS is a low-power methodology designed to dynamically adjust the ICs' working frequency and supply voltage to the actual application needs. This methodology is widely used to achieve the system trade-off between power consumption and required performance [JHO<sup>+</sup>11] [GFS<sup>+</sup>10]. DVFS has shown the potential for significant power and energy saving in many systems components, such as processor cores, memory systems, or interconnects. The DVFS approach can

adjust the supply voltage and frequency to the bare optimal necessary to meet the system requirements. However, the operating system frequency and supply voltage will have a significant influence on the system's SEU susceptibility. According to existing research, voltage scaling has a direct and negative influence on system dependability [ZMM04] [YWL<sup>+</sup>08], and frequency scaling can boost system reliability in electronic circuits [NYS<sup>+</sup>05]. There are several existing research studies that evaluate system reliability methodologies with respect to voltage and frequency scaling [ZSW00] [ZMMS04] [FSW<sup>+</sup>11]. In this thesis, a widely used accurate soft system reliability analytic module [FSW<sup>+</sup>11] is primarily employed as a baseline to accurately analyze system reliability fluctuation under dynamic voltage and frequency modifications.

### Effect of frequency on system reliability

As frequency increases, so does the total system SER. Furthermore, as compared to sequential pipeline utilization, register files and caches are the primary contributors to system SERs [IFF04]. Because pipeline timing contributes less to total SER than memory and combination logics [GSZ09], when the frequency is dropped, the overall SER declines linearly, and its slope is dependent on the system's sequential elements. As a result, Firouzi et al. [FSW<sup>+</sup>11] suggested a soft error rate model proportional to frequency scaling with a constant supply voltage:

$$SER(f) = SER_0 * \frac{f}{f_{max}} \quad (6.9)$$

where  $f$  is the operating frequency,  $f_{max}$  is the maximum frequency of the target system, and  $SER_0$  is the coefficient related to technology and represents the average soft error rate corresponding to maximum frequency and maximum voltage, i.e.  $SER_0 = SER(f_{max}, v_{max})$ .

As frequency changes, the system performance,  $Pe$ , of the system will be affected accordingly:

$$Pe(f) = Pe_0 * \frac{f}{f_{max}} \quad (6.10)$$

where the  $Pe_0$  is the system performance corresponding to maximum frequency and voltage.

Furthermore, lowering the system frequency can reduce the switching rate of the transistors, lowering the core's dynamic power consumption. The following equation can be used to compute the chip's dynamic power consumption [MIL15]:

$$P_{dp} = C * V^2 * F \quad (6.11)$$

where  $P_{dp}$  is the dynamic power consumption of the target system,  $C$  is the overall capacitance of the system,  $V$  is the supply voltage, and  $F$  is the target system frequency. Moreover, the overall power consumption includes also the leakage power consumption, which is mainly caused by subthreshold leakage current  $I_{subn}$ , and can be expressed as [MIL15]:

$$P_{lp} = V * I_{subn} \quad (6.12)$$

where  $P_{lp}$  is the leakage power consumption of the target system, and the  $V$  is the supply voltage. Therefore, the total system power consumption  $P_{dvfs}$  is:

$$P_{dvfs} = P_{dp} + P_{lp} = C * V^2 * F + V * I_{subn} \quad (6.13)$$

Since this thesis mainly focuses on the impact of DVFS technology on system reliability and does not seek accurate power consumption variation analysis, for the sake of simplicity, in the following system power consumption analysis, the focus is on dynamic power consumption. Since the term  $C * V^2$  is constant during the frequency scaling. Thus, the system power consumption for frequency scaling can be obtained approximately accordingly:

$$P(f) = P_0 * \frac{f}{f_{max}} \quad (6.14)$$

where the  $P_0$  is the system power consumption corresponding to maximum frequency and voltage.

### Effect of supply voltage on system reliability

According to [HSW00], the SER of a circuit node can be represented as exponentially related to its critical charge:

$$SER \propto F * A * e^{-\frac{Q_{crit}}{Q_s}} \quad (6.15)$$

where  $Q_{crit}$  is the minimum charge required for inducing soft error,  $A$  is the area of the sensitive area of the target system,  $Q_s$  is the charge collection efficiency of the device, and  $F$  is the radiation intensity. For the sake of simplicity, the total SER of the system can be assumed to be the sum of all circuit node SER values.

Moreover, the critical charge,  $Q_{crit}$  can be impacted by several factors [DNL08], including supply voltage, threshold voltage, transistor channel length, etc. The critical charge,  $Q_{crit}$  is proportional to the supply voltage [SML<sup>+</sup>01]. Therefore, the effects of  $Q_{crit}$  for voltage scaling are considered, and the other parameters are fixed, the relation between soft error rate and supply voltage when the frequency is constant is [FSW<sup>+</sup>11]:

$$SER(v) = SER_0 * e^{C_0(v-v_{max})} \quad (6.16)$$

where  $SER_0$  represents the average soft error rate, corresponding to maximum frequency and voltage,  $v$  is the operating voltage,  $v_{max}$  is the maximum voltage of the target system, and  $C_0$  is a fitting parameter related to the technology as well as the topology [FSW<sup>+</sup>11]. The last parameter captures nonlinear effects, which can be collected from experimental results.

The variation in power consumption caused by frequency scaling mainly affects the dynamic power of the system. On the other hand, voltage scaling would affect the target system's dynamic power and leakage power consumption, as shown in Eq. (6.13). For the sake of simplicity, only the dynamic power consumption is considered in the following analysis. Thus, according to Eq. (6.11), the approximate power consumption variation for the system with voltage scaling is:

$$P(v) = P_0 * \left(\frac{v}{v_{max}}\right)^2 \quad (6.17)$$

where  $P_0$  is the system power consumption corresponding to maximum frequency and voltage.

Moreover, the voltage can significantly affect the temperature of the chip. At low temperatures, transistor performance might suffer, and at a certain point, the system performance degradation due to heating outweighs the possible improvements from higher supply voltages. For the sake of simplicity, the system performance is considered constant when voltage scaling is applied.

### Reliability modelling of concurrent scaling of voltage and frequency

In the above analysis, the introduced equations for  $SER(f)$  (i.e. Eq. 6.9) and  $SER(v)$  (i.e. Eq. 6.16) compute the SER variation of a circuit proportional to  $SER_0 = SER(f_{max}, v_{max})$ . Moreover, the  $SER(f)$  is the system SER with varying frequencies when the supply voltage is constant, and the  $SER(v)$  represents SER for a system with voltage scaling and with a fixed frequency. Since  $SER(f)$  and  $SER(v)$  are orthogonal, and the effects of frequency and voltage are both proportional to  $SER_0$ , by using the independent probability rule, the system SER with concurrent voltage and frequency scaling normalized to  $SER_0$  is [FSW<sup>+</sup>11]:

$$SER(f, v) = \frac{f}{f_{max}} * e^{C_0(v-v_{max})} \quad (6.18)$$

Similar to the above analysis, according to Eq. (6.11), the approximation of the system power consumption with concurrent voltage and frequency scaling normalized to  $P_0$  is:

$$P(f, v) = \frac{f}{f_{max}} * \left(\frac{v}{v_{max}}\right)^2 \quad (6.19)$$

Since the impact of frequency changes on performance is mainly considered in this thesis, the system performance normalized to  $Pe_0$  is:

$$Pe(f, v) = \frac{f}{f_{max}} \quad (6.20)$$

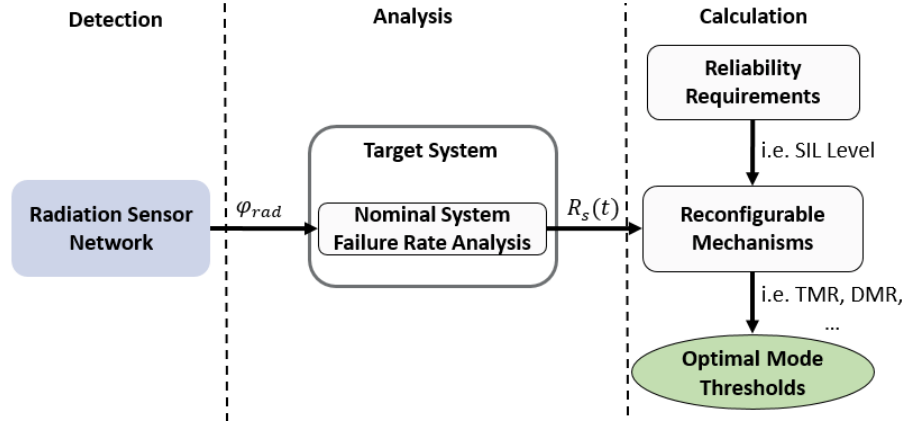
In the subsequent sections, the theoretical analysis of how the proposed SAFR method can be applied for optimal voltage and frequency selection is presented based on the above DVFS reliability equation.

### 6.3 Optimal Mode Selection

This section analyses the optimal mode selection method in the proposed SAFR system with core-level NMR and DVFS methods. The general target is to achieve the dynamic selection of system operating modes under variable working environments during run-time, thus, delivering the dynamic trade-off between reliability, performance, and power consumption in real-time. As shown in Figure 6.2, the proposed optimal mode selection is based on a series of pre-defined Look-Up Tables (LUT), which store the reliability threshold of all possible operating modes under



various reliability requirements and radiation conditions. In this section, the method for obtaining the LUT values is introduced.



**Figure 6.10:** Decision flow of the reconfigurable system optimal mode selection.

In Figure 6.10, the pre-defined optimal mode selection LUT decision flow is illustrated. The proposed flow is intended to be used in the self-adaptive platform and contains three main phases: radiation environment detection, system design-phases analysis, and optimal mode threshold calculation. In Chapters 4 and 5, radiation environment analysis is discussed, with the detected and predicted real-time in-flight SEU rate as output. During the detection phases, as shown in Figure 6.10, the information of the cross-section of the target SRAM detector  $\sigma_{SRAM}$  could be obtained through the irradiation experiment, and consequently, the predicted particle flux  $\varphi_{PRED}$  for the time period T (T = 1 hour) can be computed as:

$$\varphi_T = \frac{SEU_T}{\sigma_{SRAM} \times T} \quad (6.21)$$

where  $SEU_T$  is the number of detected or predicted upsets in the target time period,  $\sigma_{SRAM}$  is the cross-section in  $\text{cm}^2$ ,  $\varphi_T$  is the particle flux in particles/ $\text{cm}^2$ .

In the analysis phase, the system failure rate  $\lambda_s$  and corresponding reliability function  $R_s(t)$  are collected. The failure rate of the target system depends on the design, operating conditions, executed tasks and in-flight particle flux. With respect to operating conditions, the main contributors are clock frequency and supply voltage affecting the failure rate. As introduced in Subsection 6.2, higher supply voltage decreases the failure rate due to higher robustness to particle strikes, while higher frequency increases the failure rate due to higher error latching probability. If the target system is tested under a pre-defined nominal flux  $\varphi_{NOM}$ , clock frequency  $f_{nom}$  and supply voltage  $V_{nom}$ , the system failure rate,  $\lambda_s$ , under the collected flux  $\varphi_T$  can be calculated as:

$$\lambda_s = \frac{\varphi_T}{\varphi_{NOM}} * \lambda_{s\_NOM}(f_{nom}, V_{nom}) \quad (6.22)$$

where  $\lambda_{s\_NOM}(f_{nom}, V_{nom})$  denotes the nominal system failure rate for a given frequency and supply voltage. The nominal failure rate can be determined either from irradiation experiments or by simulation / analytical evaluation. If the nominal failure rate is obtained for different operating conditions, a LUT can be used to store these values, and the table readout can be employed to obtain the appropriate value for each operating mode. Moreover, from equations (6.21) and (6.22), under certain frequency, supply voltage and temperature, the  $\lambda_s$  is linearly proportional to  $SEU_T$ :

$$\lambda_s = K_s * \frac{SEU_T}{T} \quad (6.23)$$

where  $K_s$  denotes the constant parameter of the system failure rate, which can be calculated from  $\sigma_{SRAM}$ ,  $\varphi_{NOM}$  and  $\lambda_{s\_NOM}(f_{nom}, V_{nom})$ . Assuming that the system reliability decreases exponentially over time  $t$ , the system reliability function  $R_s(t)$  can be expressed as:

$$R_s(t) = e^{-\lambda_s t} = e^{-K_s SEU_T} \quad (6.24)$$

After obtaining the baseline system reliability function, it is possible to determine the optimal mode threshold from the possible system dynamic mechanisms candidates and the system reliability requirements.

### 6.3.1 Mode selection for core-level NMR

As introduced in Subsection 6.2.1, the reliability of the available NMR system with a majority voter is generally based on the binomial theorem. Combining core-level NMR operation mode reliability function Eq. (6.5)-(6.8) with the above general system reliability function with respect to  $SEU_T$  (i.e. Eq. (6.24)), the following reliability function of the exemplified quad-core system can be obtained:

$$R_{high-performance}(t) = R_{de-stress}(t) = e^{-K_c SEU_T} \quad (6.25)$$

$$R_{DMR}(t) = e^{-2K_c SEU_T} \quad (6.26)$$

$$R_{TMR}(t) = 3e^{-2K_c SEU_T} - 2e^{-3K_c SEU_T} \quad (6.27)$$

$$R_{QMR}(t) = P_d 6e^{-2K_c SEU_T} (1 - 2e^{-K_c SEU_T} + e^{-2K_c SEU_T}) + 4e^{-3K_c SEU_T} - 3e^{-4K_c SEU_T} \quad (6.28)$$

where the  $K_c$  denotes the constant parameter of core failure rate and  $SEU_T$  is the detected or predicted SEU rate from the SAFR system sensor network, and  $P_d$  denotes the probability that one error core output is different from another in the QMR mode.

The optimal operation mode should be determined from the pre-defined reliability requirements. As introduced in Section 6.1, the IEC 61508 standard [IEC05] for high-reliability systems proposes four SIL, with SIL 4 being the most reliable and SIL 1 being the least. The Probability of Failures per Hour (PFH) requirements are determined for different SIL levels, as shown in Table 6.1. The PFH of the system can be calculated as:

$$PFH_s = 1 - R_s(t), \quad t = 1 \text{ h} \quad (6.29)$$

Therefore, by combining Eq. (6.25)-(6.28) with Eq. (6.29), the reliability thresholds of corresponding NMR modes in each SIL level with respect to the SEU rate can be determined. As an example, the calculation flow of the thresholds is shown, which is based on the following two following assumptions:

- the reliability requirement of the target system is to guarantee a minimum of SIL 1 level
- the reliability of a single core can guarantee at least SIL 1 level with no-hardening mechanisms during background conditions (i.e. non-SPE period)

**Table 6.1:** Probability of Failure per Hour (PFH) for different Safety Integrity Levels (SILs) as defined in the IEC 61508 standard [IEC05].

SIL	PFH
1	$10^{-5} \sim 10^{-6}$
2	$10^{-6} \sim 10^{-7}$
3	$10^{-7} \sim 10^{-8}$
4	$10^{-8} \sim 10^{-9}$

The operation mode thresholds are determined by the value of  $\frac{SEU_T}{SEU_0}$ , where the  $SEU_0$  is the background baseline SEU rate and can be obtained by linear fitting to the detected hourly SEU average during the non-SPE period. Therefore, according to the above assumption, during the background condition, the following PFH with the baseline of SIL 1 can be collected:

$$1 - e^{-K_c} = 10^{-6} \quad (6.30)$$

Moreover, as introduced in Section 6.2.1, the DMR mode always has lower reliability, but this mode can still perform single error detection. As introduced in Section 5.3.3, based on the analysis of all the selected SPEs in the solar cycle 24, the target SRAM SEU rates for the first detected instance after solar event explosive is usually 1.5~6 times higher than the background condition. Therefore, as an example, the trigger threshold for the DMR mode is set to  $3 * 10^{-6}$  of PFH, which is three times higher than the background condition (i.e. SIL1 with PFH =  $10^{-6}$ ). Thus, the corresponding DMR mode triggering threshold can be calculated the Eq. (6.25) and the pre-defined threshold:

$$1 - e^{-K_c \frac{SEU_{DMR}}{SEU_0}} = 3 * 10^{-6} \quad (6.31)$$

Moreover, the TMR mode should be activated when the above DMR mode cannot guarantee the SIL 1 level reliability requirement. Thus, the TMR mode triggering threshold can be obtained from Eq. (6.26) and Table 6.1:

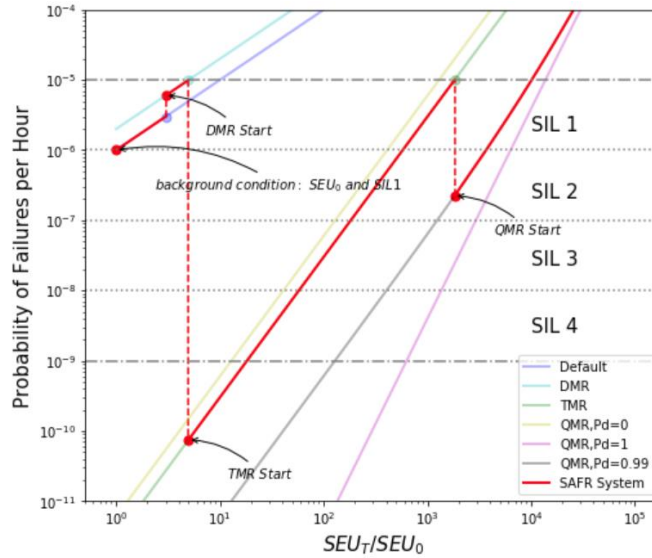
$$1 - e^{-2K_c \frac{SEU_{TMR}}{SEU_o}} = 10^{-5} \quad (6.32)$$

Similarly, as introduced in Section 6.2.1, the QMR mode would be triggered when the TMR mode cannot guarantee the requirements. Thus, the corresponding QMR mode triggering threshold can be collected from Eq. (6.27) and Table 6.1:

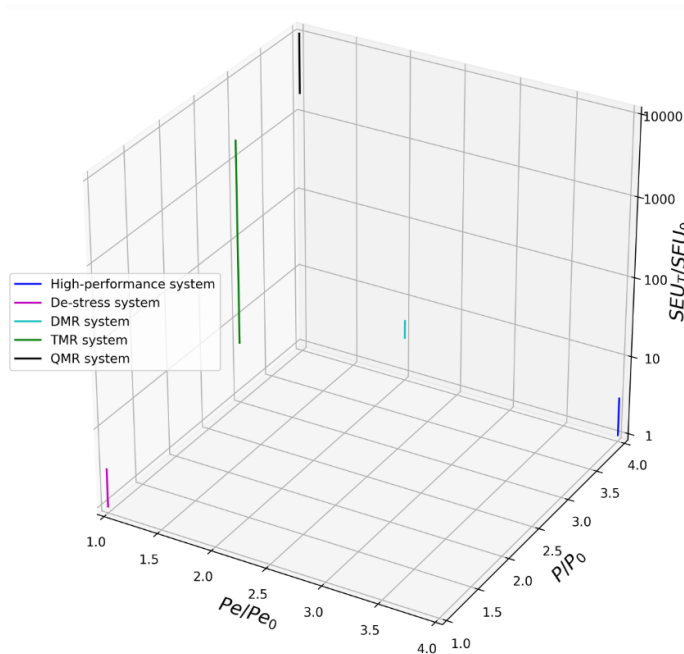
$$3e^{-2K_c \frac{SEU_{QMR}}{SEU_o}} - 2e^{-3K_c \frac{SEU_{QMR}}{SEU_o}} = 10^{-5} \quad (6.33)$$

Therefore, the corresponding core-level NMR mode trigger thresholds can be determined from Eq. (6.30)-(6.33). As a result, as shown in Figure 6.11, no core-level NMR is required when  $\frac{SEU_T}{SEU_o} < 3$ ; if  $3 \leq \frac{SEU_T}{SEU_o} < 5$ , the DMR mode can be deployed, which enables effective fault detection; if  $5 \leq \frac{SEU_T}{SEU_o} < 1830$ , the TMR is needed to keep the system SIL 1 compliant; if  $\frac{SEU_T}{SEU_o} \geq 1830$ , the QMR may need to deploy. As introduced in Section 6.2.1, for an NMR voter to produce the correct output, the number of correct cores should be greater than the number of erroneous cores providing the same wrong output signals. Thus, if  $P_d > 0.5$ , QMR system can have higher reliability than TMR, and the QMR mode should be activated. Moreover, as an example, if  $\frac{SEU_T}{SEU_o} \geq 10120$ , the QMR system with  $P_d = 0.99$  cannot meet the reliability requirement anymore.

Figure 6.12 presents the system performance and power consumption variation with respect to the  $\frac{SEU_T}{SEU_o}$  in the above core-level NMR example. The normalized baseline performance (i.e.  $Pe = 1$ ) and power consumption (i.e.  $P = 1$ ) correspond to the single-core system, as introduced in Section 6.2.1. It can be seen from the figure that as the  $\frac{SEU_T}{SEU_o}$  increases (corresponding to the increase of high energy particle flux), different fault-tolerant operation modes can be activated with the different overhead to the system performance or power consumption.



**Figure 6.11:** Example of SAFR optimal mode selection for the core-level NMR method in a quad-core system.  $SEU_T$  is in-flight hourly SEUs from the sensor network;  $SEU_0$  is average hourly SEUs during background conditions. The reference normalized value is obtained from the high-performance or de-stress mode of the quad-core system.



**Figure 6.12:** Example of system performance and power consumption with respect to the  $\frac{SEU_T}{SEU_0}$  in the SAFR optimal mode selection of the quad-core system.  $Pe$  and  $P$  represent system performance and power consumption, respectively.  $Pe_0$  and  $P_0$  are normalized baseline performance and power consumption, corresponding to the values for the single-core system.

### 6.3.2 Mode selection on DVFS

There have been many research strategies for DVFS [FCB<sup>+</sup>19] [MIL15] [RHM<sup>+</sup>17], but most of these works have focused on reducing power consumption. Contrary to related studies, this thesis mainly evaluates DVFS in the context of improving the system's reliability. As introduced in Subsection 6.2.2, the system reliability, performance and power consumption have a different dependence on the system supply voltage and frequency:

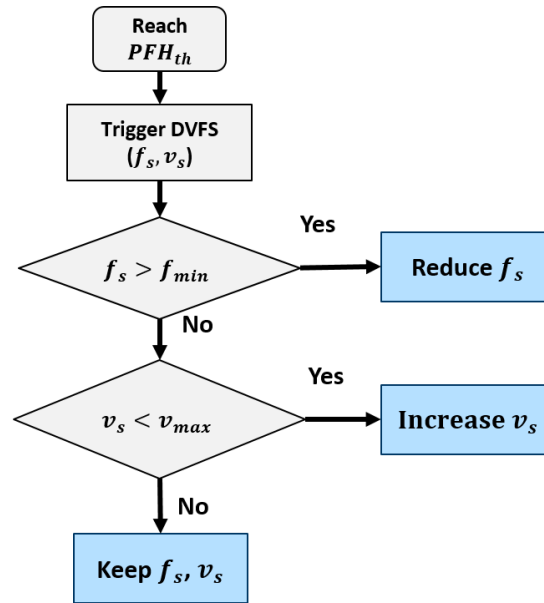
- Reliability,  $SER \propto (f, e^{C_0 v})$
- Performance,  $Pe \propto (f)$
- Power consumption,  $P \propto (f, v^2)$

By combining Eq. (6.18) and (6.23), the system failure rate with DVFS can be obtained as:

$$\lambda_{DVFS}(f, v) = K_{sm} * \frac{SEUT}{T} * \frac{f}{f_{max}} * e^{C_0(v-v_{max})} \quad (6.34)$$

where  $K_{sm}$  denotes the constant parameter corresponding to the system failure rate, which can be calculated from  $\sigma_{SRAM}$ ,  $\varphi_{NOM}$  and  $\lambda_{s\_max}(f_{max}, V_{max})$ . Therefore, the corresponding system reliability function with DVFS can be expressed as:

$$R_{DVFS}(t, f, v) = e^{-\lambda_{DVFS}(f, v)t} = e^{-K_{sm} * SEUT * \frac{f}{f_{max}} * e^{C_0(v-v_{max})}} \quad (6.35)$$



**Figure 6.13:** Decision flow of the proposed DVFS adjustment strategy to increase system reliability.  $PFH_{th}$  is the pre-defined DVFS trigger threshold with SIL standard.  $(f_s, v_s)$  are the system frequency and voltage.

The above evaluation equations can be used to perform the rough estimation of system reliability variation with respect to the supply voltage, frequency, and radiation environment. Therefore, it is possible to estimate the influence of DVFS on the system reliability and the corresponding requirements for SEU mitigation. Moreover, as mentioned in Subsection 6.2.2, when the DVFS in the setting of  $(f_{max}, v_{min})$ , the reliability of the target system is supposed to be the lowest. Therefore, according to the detected and predicted SEU rate,  $SEU_T$ , and the background baseline SEU rate,  $SEU_0$ , it is possible to evaluate when the DVFS adjustment needs to be triggered to improve system reliability, as shown in Figure 6.13. Since the baseline reliability of the system comes from  $(f_{max}, v_{min})$ , reducing the frequency could not only increase reliability but also reduce power consumption. Therefore, the proposed strategy is intended to perform the frequency scaling first until the lower limit of performance requirement is reached (i.e.  $f_{min}$ ), and then, increasing the supply voltage would be the option.

By combining Eq. (6.35) with (6.29), the adjustment equations of supply frequency and voltage for keeping the system reliability stable are:

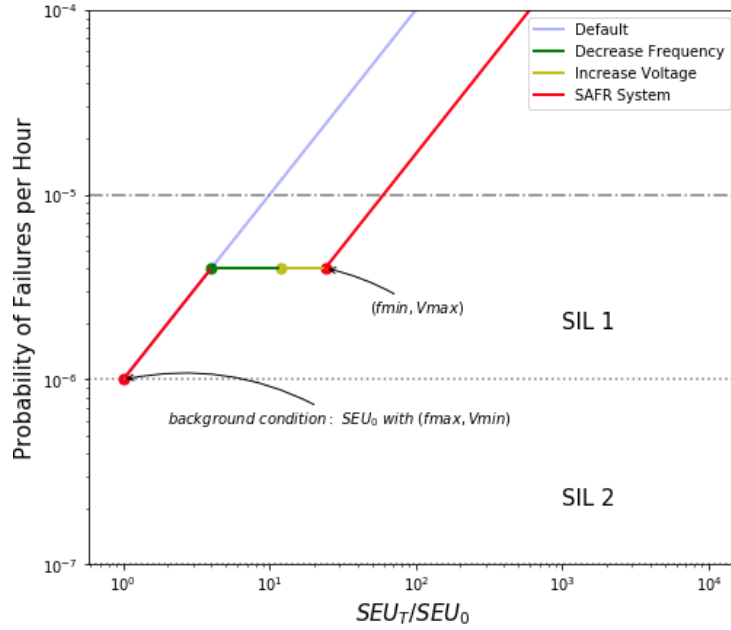
$$f_s = \frac{-\ln(1-PFH_{th}) * f_{max}}{e^{C_0(v_s - v_{max})} * K_{sm} * SEU_T} \quad (6.36)$$

$$v_s = \frac{\ln\left(\frac{-\ln(1-PFH_{th}) * f_{max}}{K_{sm} * SEU_T * f_s}\right)}{C_0} + v_{max} \quad (6.37)$$

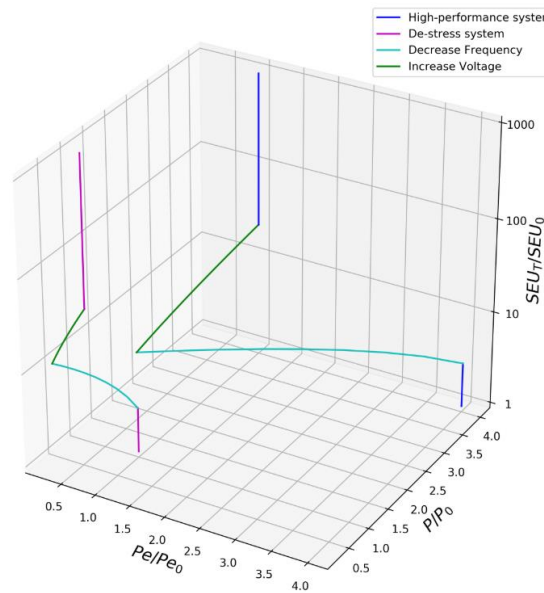
where  $PFH_{th}$  is the pre-defined threshold for triggering the DVFS approach. Since this thesis includes only a theoretical evaluation of the DVFS approach in the context of reliability, the detailed analysis of the parameters in Eq. (6.36) and (6.37) would be the future work. In order to show the SAFR system with the DVFS method, as an example, the following assumptions are applied:

- the reliability requirement of the target system is to guarantee a minimum of SIL 1 level
- in the case of DVFS combination of  $(f_{max}, v_{min})$ , the corresponding system reliability can guarantee at least SIL 1 level during background conditions (i.e. non-SPE period)
- In the target system,  $\frac{f_{max}}{f_{min}} = 4, v_{min} = 0.7V, v_{max} = 1.4V$  with  $PFH_{th} = 3 * 10^{-6}$

Figure 6.14 presents the DVFS optimal mode selection method for the proposed SAFR system based on the above example assumptions and Eq. (6.36) (6.37). As a result, the  $(f_{max}, v_{min})$  combination can be used when  $\frac{SEU_T}{SEU_0} < 3$ ; if  $3 \leq \frac{SEU_T}{SEU_0} < 12$ , the frequency scaling methods need to be deployed to keep the reliability of the system, until  $f_{min}$  is reached; if  $12 \leq \frac{SEU_T}{SEU_0} < 24$ , the power supply voltage needs to be gradually increased until it reaches  $v_{max}$ . When the system reaches  $(f_{min}, v_{max})$ , the DVFS method can no longer be used to increase the system's reliability. It is worth noting that this DVFS method is only for system reliability adjustments. The possible adjustment range for system voltage or frequency could be more limited in practical DVFS applications, limited by real-time system performance or power consumption requirements.



**Figure 6.14:** Example of SAFR optimal mode selection for the DVFS method in a system.  $SEU_T$  is the in-flight hourly SEUs from the sensor network;  $SEU_0$  is the average hourly SEUs during background conditions. The default operation can be the high-performance or de-stress operation mode.



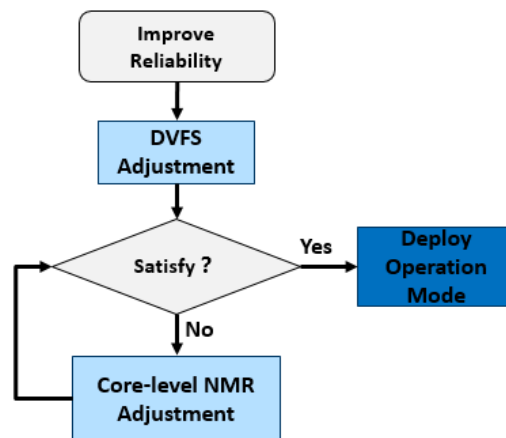
**Figure 6.15:** Example of system performance and power consumption with respect to  $\frac{SEU_T}{SEU_0}$  with the DVFS method. The baseline of performance ( $Pe_0$ ) and power consumption ( $P_0$ ) is the corresponding value for a single core with  $(f_{max}, v_{min})$ .



By applying DVFS system performance and power consumption Eq. (6.19) and (6.20) to the above example, system performance and power consumption variation with respect to the  $\frac{SEU_T}{SEU_0}$  can be obtained, as shown in Figure 6.15. The reference performance (i.e.  $Pe = 1$ ) and power consumption (i.e.  $P = 1$ ) correspond to the features of a single core with  $(f_{max}, v_{min})$ . It can be seen from the figure that as  $\frac{SEU_T}{SEU_0}$  increases, the system frequency and the supply voltage should be adjusted accordingly. As a result, as the system's frequency is reduced, the system's performance will also be reduced. Similarly, the system power consumption would continue to vary within a certain range by adjusting the supply voltage and frequency. This example mainly shows the changes in system power consumption and performance under the condition that DVFS adjusts the system's reliability. However, the system can achieve higher performance or lower energy consumption by means of traditional DVFS adjustment during the background radiation condition.

### 6.3.3 Concurrent selection of core-level NMR and DVFS

The above analysis shows that the impact of the core-level NMR and DVFS on system reliability, power consumption, and performance have different characteristics. The core-level NMR can improve the system's reliability, especially for TMR and QMR, but this improvement is coarse-grained, and the fine-grained reliability adjustment cannot be achieved. Moreover, it also affects the parallel processing capacity of multiprocessing systems. On the other hand, DVFS can achieve fine-grained adjustment of system reliability. However, the reliability adjustment range with DVFS is limited and may not suit scenarios requiring a significant increase in system reliability. Therefore, these two methods can be combined to address more diverse needs and scenarios.



**Figure 6.16:** Decision flow of the core-level NMR and DVFS methods to increase system reliability in the proposed SAFR system.

The optimal mode selection strategy for concurrent core-level NMR and DVFS is shown in Figure 6.16. The DVFS method is utilized for fine-grained reliability adjustments, thus, ensuring the system reliability from small SEU fluctuations. Moreover, as introduced in Section 6.1, the proposed SAFR system can detect and predict the radiation environment. Thus, the adjustment of DVFS needs to meet actual (based on detected SEU rate) and future (based on predicted SEU rate) reliability at the same time. When DVFS cannot meet the reliability requirements (i.e. reach  $(f_{min}, v_{max})$ ), and the system's reliability needs to be significantly improved, the core-level NMR method should be adopted. As shown in Figure 6.9, the system reliability of DMR mode is even lower than in de-stress mode, and its primary function is to detect errors. Therefore, the DMR model is no longer considered here.

The following assumptions, which are the same hypothetical templates in Sections 6.3.1 and 6.3.2, are used as an example to show optimal mode concurrent selection with the core-level NMR and DVFS methods in the proposed SAFR system:

- the reliability of the target system has to guarantee a minimum of SIL 1 level
- in the case of DVFS combination of  $(f_{max}, v_{min})$  with the de-stress or high-performance mode, the corresponding system reliability can guarantee at least SIL 1 level during background conditions (i.e. non-SPE period)
- In the target system,  $\frac{f_{max}}{f_{min}} = 4, v_{min} = 0.7V, v_{max} = 1.4V$  with  $PFH_{th} = 3 * 10^{-6}$

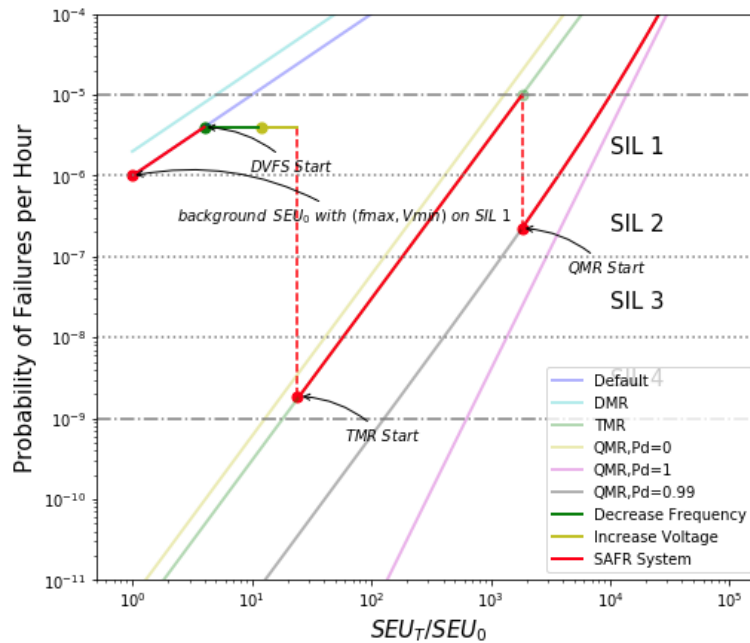
Figure 6.17 presents the optimal mode selection based on IEC 61508 standard with core-level NMR and DVFS. This example is based on the above provided three assumptions, and the corresponding mode selection thresholds are as follows:

- when  $\frac{SEU_T}{SEU_o} < 3$ , no need for additional protection for the target system, and the multiprocessor can be set to de-stress or high-performance mode with  $(f_{max}, v_{min})$ .
- when  $3 \leq \frac{SEU_T}{SEU_o} < 12$ , as introduced in Subsection 6.3.2, the frequency scaling method needs to be deployed. The corresponding system reliability, performance and power consumption can be referred to Eq. (6.9), (6.10) and (6.14).
- when  $12 \leq \frac{SEU_T}{SEU_o} < 24$ , the system supply voltage needs to be increased, which is detailed in Subsection 6.3.2, with the corresponding system reliability and power consumption variation Eq. (6.16) and (6.17), respectively.
- when  $24 \leq \frac{SEU_T}{SEU_o} < 1830$ , the core-level TMR mode should be triggered. Since the TMR mode can significantly improve the system's reliability, the DVFS method may not be necessary for increasing system reliability. Therefore, the DVFS can be reset to the baseline setting or used for system energy saving during this condition.
- when  $\frac{\mu}{\mu_o} \geq 1830$ , the QMR may need to be deployed. Same to the above TMR case, the DVFS can also be used for energy saving during this phase.

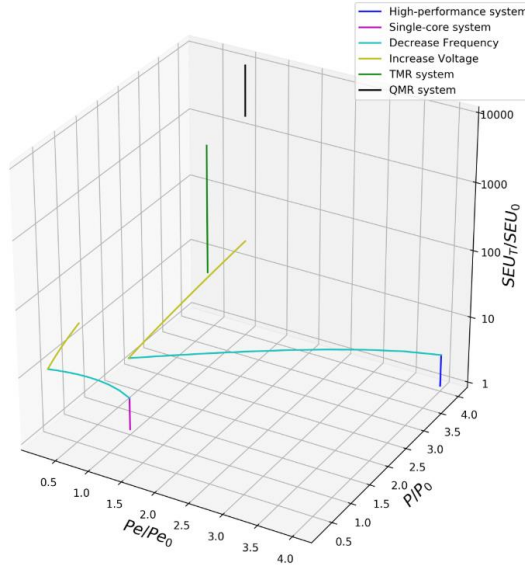
As mentioned in the above introduction, during TMR and QMR phases, the DVFS method can increase the system's stability further. Since these two methods are independent of each other, using  $\lambda_{DVFS}(f, v)$  into the core-level NMR reliability function, the corresponding reliability function combining both two methods can be obtained. As an example, Eq. (6.38) shows the reliability function by concurrent TMR with DVFS from Eq. (6.27) and (6.34). However, as introduced in Section 6.4, after analyzing all solar events in solar cycle 24, it can be found that core-level TMR is sufficient for most solar events, and only a few very large SPEs need to trigger QMR. Thus, using DVFS to enhance reliability during TMR and QMR may not be necessary.

$$R_{TMR_{DVFS}}(t, f, v) = 3e^{-2K_{sm} * SEU_T * \frac{f}{f_{max}} * e^{C_0(v-v_{max})}} - 2e^{-3K_{sm} * SEU_T * \frac{f}{f_{max}} * e^{C_0(v-v_{max})}} \quad (6.36)$$

Based on the analysis results in Sections 6.3.1 and 6.3.2, Figure 6.18 presents the example of system performance and power consumption variation with respect to the  $\frac{SEU_T}{SEU_0}$  of the core-level NMR and DVFS methods. The baseline performance (i.e.  $Pe = 1$ ) and power consumption (i.e.  $P = 1$ ) correspond to single-core behaviour with  $(f_{max}, v_{min})$ .



**Figure 6.17:** Example of SAFR optimal mode selection for a quad-core system's core-level NMR and DVFS methods.  $SEU_T$  is the in-flight SEUs from the sensor network;  $SEU_0$  is the average SEUs during background conditions. The default value is obtained from the cores in high-performance or de-stress operation mode with  $(f_{max}, v_{min})$ .



**Figure 6.18:** Example of system performance and power consumption with respect to the  $\frac{SEU_T}{SEU_0}$  in the SAFR optimal mode selection of the quad-core system with core-level NMR and DVFS method. The baseline of performance ( $P_{e_0}$ ) and power consumption ( $P_0$ ) is the corresponding value for a single core with  $(f_{max}, v_{min})$ .

#### 6.4 Analysis of Results

Since the proposed SAFR system is intended to dynamically protect the target system from harsh radiation conditions, especially from SPEs, it is essential to evaluate the system behaviour with various SPEs. As introduced in Chapter 5, all 36 SPEs during solar cycle 24 (2008-2019) are analyzed, and the corresponding hourly SEU rates of a 2 Gbit 65 nm SRAM are calculated from space flux databases. Therefore, these SEU rate data are used here to analyze the optimal model selection strategy of the proposed SAFR system on the exemplary quad-core system with core-level NMR and DVFS methods.

**Table 6.2:** Simulated Non-SPE period SEU monitor network behaviour of the target 2 Gbit SRAM during solar cycle 24.

	SEU Rate (Upsets/bits/day)	SEU Count (Upsets/hour/2Gbit)
<b>Minimum</b>	$2.58 * 10^{-7}$	23
<b>25%</b>	$3.84 * 10^{-7}$	34
<b>50%</b>	$4.54 * 10^{-7}$	41
<b>75%</b>	$5.74 * 10^{-7}$	51
<b>Maximum</b>	$7.19 * 10^{-7}$	64

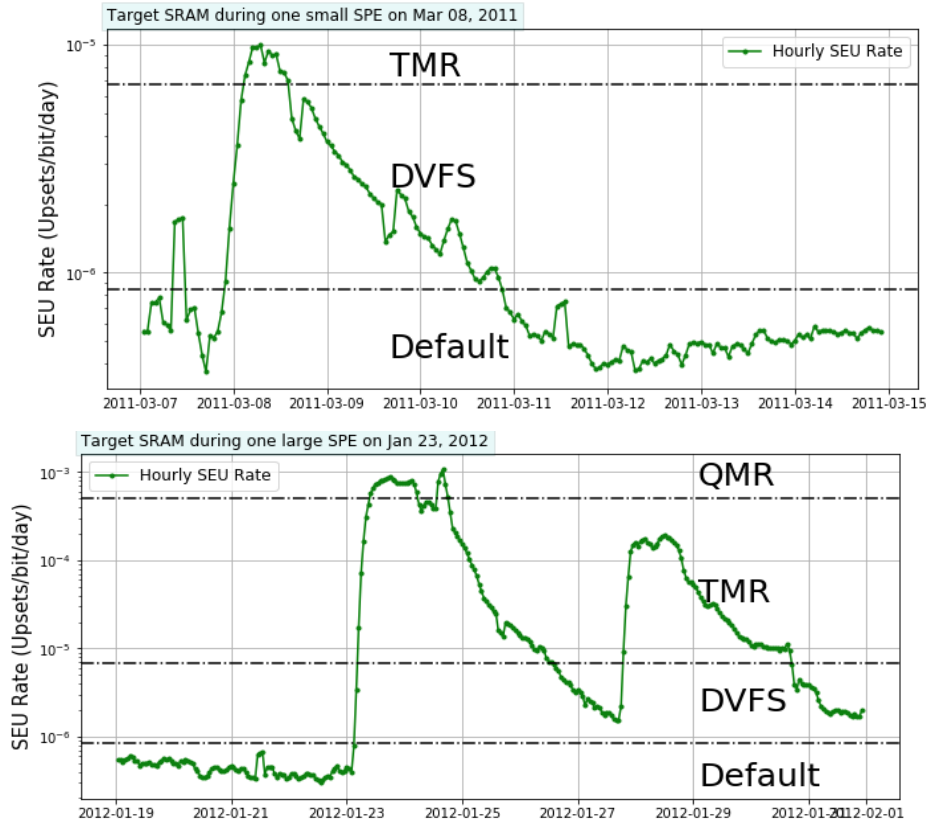
**Table 6.3:** Simulated SPE period SEU monitor network behaviour of the target 2 Gbit SRAM during solar cycle 24.  $SEU_0$  is the baseline SEU rate, which is set to  $2.80 * 10^{-7}$  upsets/bits/day.

SEU Rate Range	Maximum SEU Rate (Upsets/bits/day)	Duration Time (Hour)	Duration Time/Year (Hour)
$<10SEU_0$	$2.78 * 10^{-6}$	1057	96.1
$10SEU_0 \sim 10^2 SEU_0$	$2.79 * 10^{-5}$	979	89.0
$10^2 SEU_0 \sim 10^3 SEU_0$	$2.78 * 10^{-4}$	273	24.8
$>10^3 SEU_0$	$1.32 * 10^{-3}$	103	9.4

Since the proposed run-time optimal mode selection method is based on the detected and predicted in-flight SEU rate,  $SEU_T$ , and the pre-defined baseline SEU rate,  $SEU_0$ , it is important to select the appropriate  $SEU_0$  for the target SRAM. Through the analysis of all collected hourly SEU data for non-SPE periods during solar cycle 24, the corresponding non-SPE periods SEU rate data of the target 2 Gbit SRAM is shown in Table 6.2. As introduced in Subsection 2.1, the variation in solar activity during a solar cycle (i.e. solar maximum and solar minimum) would affect the space baseline radiation environment. Moreover, as mentioned in Subsection 5.3.3, A few hours before the start of SPE, the SEU rates could begin to increase. Therefore, the hourly SEU rates during non-period are not constant and would vary within a specific range. Generally, it is possible to adjust the baseline upset rate from a linear fit to the detected hourly SEU data. In this section, as an example, the baseline SEU rate,  $SEU_0$ , is set to  $2.80 * 10^{-7}$  upsets/bits/day (i.e. SEU count to 25 upsets/hour/2Gbit), which is a suitable value for most situations.

Through a comprehensive analysis of the target 36 solar events, it can be found that the SPEs occurred for a total of about 2,400 hours. Moreover, the majority of SPEs are small and medium-sized events, with the peak hourly SEU rates lower than 1000 times the background SEU rate. However, a few of these solar events could cause the peak hourly SEU rates to increase thousands of times than the background condition. Table 6.3 presents the analysis results of the SEU rates of target SRAM during SPE periods on solar cycle 24, as introduced in Chapter 5. Based on the baseline SEU rate,  $SEU_0$ , the detected SEU rate is divided into four categories in increments of 10 times. The duration time represents the total number of hours the corresponding SEU rate category occurs in the entire solar cycle 24. Since each solar cycle has 11 years, for the convenience of analysis, the "duration time/year" is used to present the average time of the corresponding SEU rate in one year. It should be noted that the result of "duration time/year" is only used for average analysis and is inconsistent with the actual SPE phenomenon, where large-scale solar events are usually concentrated in a few years of each solar cycle. In addition, since there are a total of 8760

hours per year, according to Table 6.3, each "average year" can be divided into 8540 hours (i.e. 97.5%) non-SPE phases and 220 hours (i.e. 2.5%) SPE phases.



**Figure 6.19:** Example of SAFR system optimal mode selection with core-level NMR and DVFS during large and small solar events, respectively. The system's default operation is the high-performance or de-stress mode.

**Table 6.4:** Example of the optimal mode selection of the SAFR system with core-level NMR and DVFS methods in a quad-core multiprocessor.  $SEU_0$  is the baseline SEU rate of the target SRAM, which is estimated as  $2.80 * 10^{-7}$  upsets/bits/day.

SEU Rate Range	Operation Mode	Duration Time/Year (Hour)
$<3SEU_0$	High-performance De-stress	8540
$3SEU_0 \sim 24SEU_0$	DVFS	142
$24SEU_0 \sim 1830SEU_0$	TMR	74
$>1830SEU_0$	QMR	4

Considering optimal mode selection threshold example of core-level NMR and DVFS methods in the quad-core multiprocessor, as introduced in Section 6.3, the corresponding behaviour of the proposed SAFR system during solar cycle 24 can be obtained, as shown in Table 6.4. According to the average duration time of each operation mode, it can be noticed that additional system reliability hardening is not required most of the time. Moreover, DVFS and TMR modes are suitable for most solar events, especially for small and medium-sized SPEs, and QMR mode can guarantee the system's reliability during the peak period of large solar events. Figure 6.19 shows the SEU rate of the proposed SAFR system during large and small solar events, respectively. Moreover, in the target solar events, the target SRAM's collected maximum hourly SEU rate is  $1.32 * 10^{-3}$  upsets/bits/day (i.e. 118122 upsets/hour/2Gbit), which is approximately 4700 times the baseline SEU rate. In the example of Section 6.3, the QMR system could maintain the system reliability level even if the SEU rate is 10120 times increased, compared to the baseline case. Therefore, the proposed core-level NMR and DVFS methods could guarantee system reliability requirements during extremely high-intensity solar events.

**Table 6.5:** Example calculation of the system power consumption and performance as well as system reliability (i.e. time of satisfying SIL 1 requirement) comparison of different modes for a quad-core multiprocessor in one year. SAFR system<sup>1</sup> and system<sup>2</sup> refer to the default operation mode during the non-SPE period with de-stress and high-performance mode, respectively.

Operation Mode	Power Consumption	Performance	Reliability
SAFR system <sup>1</sup>	8849 <i>P</i>	8689 <i>Pe</i>	100%
SAFR system <sup>2</sup>	34682 <i>P</i>	34522 <i>Pe</i>	100%
De-stress	8760 <i>P</i>	8760 <i>Pe</i>	98.6%
DMR	17520 <i>P</i>	8760 <i>Pe</i>	97.9%
TMR	26280 <i>P</i>	8760 <i>Pe</i>	99.95%
QMR	35040 <i>P</i>	8760 <i>Pe</i>	100%
High-performance	35040 <i>P</i>	35040 <i>Pe</i>	98.6%

Moreover, besides triggering on-demand operation modes, the proposed SAFR system can also effectively reduce the total system power consumption and maintain performance. According to the obtained operating time for each mode in one average year mode, as shown in Table 6.4, and corresponding operating modes behaviours, as detailed in 6.2, these modes' power consumption and performance can be obtained. Table 6.5 presents system power consumption and performance as well as system reliability of the above SAFR optimal mode selection example for a quad-core multiprocessor in one average year. The reliability column represents how many percent of the time in one average year, the corresponding system can meet the example reliability requirements

in Section 6.3 (i.e. the reliability of the target system has to guarantee a minimum of SIL 1 level, and the system reliability can guarantee at least SIL 1 level during the background condition). Moreover, since the default operation mode during the non-SPE period could be the de-stress or high-performance case, thus, both these two cases SAFR variants (i.e. SAFR system<sup>1</sup> and system<sup>2</sup>) are shown in Table 6.5. The reference system power consumption (i.e.  $P = 1$ ) and performance (i.e.  $Pe = 1$ ) correspond to a single core behaviour in one hour with  $(f_{max}, v_{min})$ . In addition, since DVFS utilization leads to continuous power consumption and performance changes, this calculation has simplified the average power consumption and performance in one hour are applied during the DVFS adjustment phase. Therefore, as an example, the cumulative system energy and performance per year for the SAFR quad-core system with default high-performance mode during one average year can be calculated as:  $P_{SAFR} = 4*8540 P + 2*142 P + 3*74 P + 4*4 P = 34682 P$ ;  $Pe_{SAFR} = 4*8540 Pe + 2*142 Pe + 1*74 Pe + 1*4 Pe = 34522 Pe$ . Comparing proposed SAFR mode switching method with the other stand-alone mode shows that the proposed system can provide adequate system reliability in any hostile radiation environment with a low penalty of performance and power consumption. When the default mode is de-stress (use of only one processor core), the power consumption of the proposed method is much lower than the other core-level NMR stand-alone mode; when the default mode is high-performance, the proposed method can provide similar system performance as the stand-alone high-performance mode.

## 6.5 Summary

This chapter presents an autonomous reconfigurable fault mitigation methodology, the SAFR system, analyzed and implemented in a quad-core multiprocessing system for reliability-critical applications. The primary function of the SAFR system is to ensure system reliability in a complex and dynamic radiation environment, thus, achieving an optimized trade-off between system reliability, power consumption, and performance. Various reconfigurable methods can be deployed on the different multiprocessing system layers, such as core-level NMR, DVFS, etc. Therefore, a multiprocessing system can form a number of operating modes with different reliability features, such as core-level TMR (lock-step) for high reliability, no core-level redundancy for high performance, etc. The proposed SAFR system can utilize the detected and predicted radiation environment data from the monitor network, as well as the pre-defined reliability requirement to determine optimal operating mode of the target multiprocessor. Therefore, a set of hardening strategies can be obtained from the SAFR method. The target multiprocessing system can automatically set different operation modes under different radiation intensity environments, thereby realizing the required system reliability with optimized power consumption and performance.

The concept, features, and implementation of different modes in the SAFR system are discussed. The mode triggering thresholds can be calculated from the corresponding mode reliability function and the required system reliability level. In this chapter, an optimal mode selection example with



core-level NMR and DVFS methods, as well as IEC 61508 standard in a quad-core multiprocessing system, is analyzed in detail. Using the proposed example, the optimal mode thresholds with the other cases can be easily obtained with the same calculation flow.

The solar events in solar cycle 24 (2008-2019) are used to evaluate the proposed SAFR system. The results show that the high-reliability modes only account for about 2.5% of each year's total operating time. Moreover, the introduced core-level NMR methods could ensure the reliability requirements of the system in all solar intensity events. Besides that, the power consumption and performance of the proposed dynamic SAFR system are better than the other static operating modes.

## Chapter 7

### Conclusion

The preceding chapters described a self-adaptive resilient operation addressing radiation-induced faults in multiprocessing systems for space applications. Mitigating radiation-induced faults is an essential need in electronic systems in radiation-harsh environments, such as space. However, due to technology scaling and the increased design complexity, mitigating radiation-induced faults in digital designs nowadays leads to even more pronounced challenges. A particular challenge is a need to address mixed-criticality requirements, which are often present in modern embedded systems. In mixed-criticality systems, the requirements of the running applications concerning reliability, performance, and power consumption are dynamically changing. Besides that, for space applications, the intensity of cosmic radiation can differ over five orders of magnitude within a few hours or days during Solar Particle Events (SPEs), resulting in several orders of magnitude variation of fault probability. However, current designers usually apply to SEE mitigation methods in a static manner (i.e., according to the worst-case condition), which is unnecessary in most cases and would induce static fault-tolerance overhead.

In order to solve the aforementioned challenges, this thesis has contributed to the present state-of-the-art methodologies by addressing the three critical features of self-adaptive resilience in processing systems: (1) Performing real-time particle detection by using existing on-chip SRAM, thus, minimizing the hardware and power overhead; (2) Prediction of SRAM SEU rate and the corresponding solar condition one hour in advance during SPEs, as well as under normal radiation conditions; (3) Self-adaptive fine-grained optimal mode selection according to the in-flight space weather and pre-defined reliability requirements, and conveniently integrated into the reconfigurable system to dynamically optimize fault mitigation methods with negligible costs. Therefore, no mitigation or mitigation with low overhead modes can be deployed in the system during the non-critical periods. On the contrary, fault tolerance mode with higher overheads can be applied dynamically during periods of high solar activities. The proposed design is aimed at highly reliable multiprocessors used in space applications, ensuring self-adaptive fault resilience,

and thereby triggering dynamic trade-offs between system reliability, power consumption, and performance during run-time.

## 7.1 Achieved Results

A complete self-adaptive resilient methodology has been proposed to facilitate dynamic fault mitigation for space applications in multiprocessing systems. The proposed methods significantly advance the existing state-of-the-art designs, especially in radiation condition monitoring, solar environmental analysis, and dynamic system implementation.

As support for detecting in-flight radiation environments, an on-chip low-cost non-standalone radiation monitor has been proposed. Due to the high radiation sensitivity of Static Random Access Memory (SRAM) blocks, the idea of the introduced monitor is to employ the standard on-chip SRAM blocks as a particle detector in parallel to its normal data storage function. The proposed monitor design extends the basic functionality of the SRAM bank by integrating an Error Detection and Correction (EDAC) module, a scrubbing module, a predefined dedicated detection flow, an Over-counting Detection Register File (ODRF), and a fault management unit on the same chip. As a result, SEUs and MCUs, as well as permanent faults in each memory word, can be accurately detected and counted. Thus, the proposed monitor can provide in-flight SRAM SEU measurement during run-time. Two implementations of the proposed monitor with external stand-alone SRAM banks and internal memory banks in a quad-core system have also been explored. Moreover, the induced area and power consumption overheads of the proposed monitor for the host system are less than 1%.

A space radiation condition predictor is implemented in real-time and autonomously to forecast the space flux variations and harsh radiation phenomena, i.e., to predict the upcoming SPEs. The detected in-flight SEU rate from the proposed SEU monitor can be used to achieve the fine-grained prediction of solar conditions during SPEs and under normal conditions. The proposed solar condition predictor method can collect the in-flight real-time SEU rate data and an offline trained machine learning model for forecasting the following SEU rate, thus, determining the corresponding solar conditions. Therefore, the upcoming flux variation, and the corresponding SPE, can be predicted based on the SEU count rate variations at least one hour in advance. 36 SPEs, which occurred in solar cycle 24 (2008-2019), are analyzed as targets, and the corresponding space proton and heavy-ion flux data are collected. As a result, over 5000 hourly SEU rate data of a 65 nm SRAM are calculated and used to train five existing machine learning regression models with the time series method. Moreover, the online learning method is also introduced, which can trigger adapting the system to completely unexpected situations which could not be predicted based on the offline training. A dedicated low-cost hardware accelerator is also customized to execute the proposed machine learning prediction model and online learning algorithm with excellent prediction accuracy.

In order to overcome the drawback of static mitigation in the complex and dynamic radiation environment, an autonomous reconfigurable fault mitigation methodology, the Self-Adaptive Fault Resilience (SAFR) system, is presented for reliability-critical applications. The system combines the proposed radiation monitor and prediction model with dynamic system mechanisms to perform real-time optimal operation mode selection. The proposed SAFR system can utilize detected and predicted radiation environment data from the monitor network and pre-defined reliability requirements to determine the optimal operating mode among the available dynamic schemes in the target system. The target of the SAFR system is to overcome the limitations of static mitigation designs and achieve a dynamic trade-off between system reliability, power consumption, and performance. Moreover, the evaluation process of the optimal mode is introduced, which can be applied to common dynamic mechanisms and reliability standards. As an example, the optimal mode selection method in a quad-core multiprocessing system with core-level N-Module Redundancy (NMR) and Dynamic Voltage Frequency Scaling (DFVS) methods, as well as IEC 61508 standard, is analyzed in detail. Furthermore, the space flux data during solar cycle 24 are applied to evaluate the proposed quad-core system. The results of the analysis show that only about 2.5% of the total operating time per year requires a high-reliability mode. Furthermore, the power consumption and performance of the proposed adaptive system outperform other static operating modes.

The results of this thesis can contribute to establishing a complete design flow for self-adaptive resilient processing systems and serve as the basis for the design of reliability-critical IC applications. To the best of our knowledge, there is currently no solution for the self-adaptive resilient system design flow of ASIC-based embedded systems, especially multiprocessing systems, which support fault detection, environment prediction, and real-time optimization of the system's operation mode. Introducing such solutions is critical for designing dynamic mix-critical embedded systems with variable radiation conditions, especially for space-borne applications.

## **7.2 Future Work**

Aside from the study objectives described in Chapter 1, there is much more research that can be done to further refine the self-adaptive resilient system design process illustrated in Figure 1.1. This section covers the important strategic directions of future work in a concise manner.

Follow-up work on on-chip monitors can be done in two directions: (1) improve the present radiation monitor; (2) create and deploy other types of on-chip monitors. The suggested radiation monitor, as the major component of the self-adaptive system, can be enhanced by detecting and analyzing more radiation effects, such as Total Ionizing Dose (TID), Single Event Transient (SET), etc. In order to do accurate intensity analysis, it is also necessary to be able to quantify the Linear Energy Transfer (LET) of radiated particles. Aside from radiation particles, many other environmental effects, such as aging, temperature, and so on, could have an impact on the system's

reliability during operation. As a result, it is also critical to evaluate the influence of other factors more thoroughly and to design and integrate appropriate monitoring.

The solar condition predictor can be improved by calibrating with more predictive models and historical space ion fluxes. Since solar cycles vary in strength, more historical solar events across different solar cycles can be analyzed to ensure a more comprehensive sample. Additionally, additional machine learning and neural network models can be applied for data analysis. On one hand, the accuracy of the forecast can be improved, and on the other hand, the space environment would be predicted by several hours in advance. In addition, the prediction accuracy can be further improved by collecting more types of data, such as the real-time measurement of particle LET, which can be used as an additional input parameter for the machine learning algorithm.

In addition to the embedded multiprocessing system implemented in this thesis, the proposed concept of the SAFR system intends to be used with more processing units, such as Graphics Processing Units (GPUs), Digital Signal Processors (DSPs), Data Processing Units (DPUs), Field Programmable Gate Arrays (FPGAs), etc. These components can change their processing behavior w.r.t. dependability by loading more dynamic methods on different system layers. Moreover, more reconfigurable methods of software, or hardware can be evaluated with the proposed system. Furthermore, besides radiation, it is also necessary to study the influence of other factors on the selection of the optimal mode, such as aging, temperature, etc.

## Bibliography

- [ACE] Advance composition explorer - solar isotope spectrometer database, URL: [http://www.srl.caltech.edu/ACE/ASC/level2/lv2DATA\\_SIS.html](http://www.srl.caltech.edu/ACE/ASC/level2/lv2DATA_SIS.html).
- [ACS<sup>+</sup>21] M. Andjelkovic, J. Chen, A. Simevski, O. Schrape, M. Krstic and R. Kraemer, "Monitoring of Particle Count Rate and LET Variations With Pulse Stretching Inverters," in *IEEE Transactions on Nuclear Science*, vol. 68, no. 8, pp. 1772-1781, Aug. 2021, doi: 10.1109/TNS.2021.3076400.
- [ACS<sup>+</sup>20] M. Andjelkovic, J. Chen, A. Simevski, Z. Stamenkovic, M. Krstic and R. Kraemer, "A Review of Particle Detectors for Space-Borne Self-Adaptive Fault-Tolerant Systems," 2020 IEEE East-West Design & Test Symposium (EWDTS), 2020, pp. 1-8, doi: 10.1109/EWDTS50664.2020.9225138.
- [AJE<sup>+</sup>17] A.J. Engell, et al., "SPRINTS: a framework for solar-driven event forecasting and research," *Space Weather* 15 (10) (2017) 1321–1346, doi: 10.1002/2017SW001660.
- [AJT97] A. J. Tylka et al., "CREME96: A Revision of the Cosmic Ray Effects on Micro-Electronics Code," in *IEEE Transactions on Nuclear Science*, vol. 44, no. 6, pp. 2150-2160, Dec. 1997, doi: 10.1109/23.659030.
- [ALB<sup>+</sup>04] A. Avizienis, J. -. Laprie, B. Randell and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," in *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11-33, Jan.-March 2004, doi: 10.1109/TDSC.2004.2.
- [ALU<sup>+</sup>12] A. Lukefahr et al., "Composite Cores: Pushing Heterogeneity Into a Core," 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture, 2012, pp. 317-328, doi: 10.1109/MICRO.2012.37.
- [APS<sup>+</sup>15] M. Andjelkovic, V. Petrovic, Z. Stamenkovic, G. Ristic, G. Jovanovic, "Circuit-level Simulation of the Single Event Transients in an On-chip Single Event Latchup Protection Switch," *Journal of Electronic Testing - Theory and Applications*, vol. 31, pp. 275 - 289, 2015, doi: 10.1007/s10836-015-5529-1.
- [ASC<sup>+</sup>22] M. Andjelkovic, A. Simevskia, J. Chen, et al., "A design concept for radiation hardened RADFET readout system for space applications," *Microprocessors and Microsystems*, Volume 90, 2022, doi: 10.1016/j.micpro.2022.104486.
- [ASII14] A. Simevski, "Architectural framework for dynamically adaptable multiprocessors regarding aging, fault tolerance, performance and power consumption," PhD dissertation, BTU Cottbus-Senftenberg, 2014.

- [AVC<sup>+</sup>19] M. Andjelkovic, M. Veleski, J. Chen, A. Simevski, M. Krstic and R. Kraemer, "A Particle Detector Based on Pulse Stretching Inverter Chain," 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2019, pp. 594-597, doi: 10.1109/ICECS46596.2019.8964644.
- [BAL<sup>+</sup>95] J. Barak, E. Adler, Y. Lifshitz and J. Levinson, "Detecting heavy ions and protons in space: single-events monitor," Eighteenth Convention of Electrical and Electronics Engineers in Israel, 1995, pp. 5.5.1/1-5.5.1/3, doi: 10.1109/EEIS.1995.514195.
- [BDH] Benchmark-dhrystone. URL: <https://github.com/sifive/benchmark-dhrystone>.
- [BDS03] J. L. Barth, C. S. Dyer and E. G. Stassinopoulos, "Space, atmospheric, and terrestrial radiation environments," in IEEE Transactions on Nuclear Science, vol. 50, no. 3, pp. 466-482, June 2003, doi: 10.1109/TNS.2003.813131.
- [BGG<sup>+</sup>98] A. Bondavalli, F. Di Giandomenico, F. Grandoni, D. Powell and C. Rabejac, "State restoration in a COTS-based N-modular architecture," Proceedings First International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '98), 1998, pp. 174-183, doi: 10.1109/ISORC.1998.666787.
- [BGT<sup>+</sup>18] R. Possamai Basstos, L. Acunha Guimaraes, F. Sill Torres, L. Fesquet, "Architectures of Bulk Built-in Current Sensors for Detection of Transient Faults in Integrated Circuits," Microelectronics Journal, Volume 71, Pages 70-79, ISSN 0026-2692, 2018, doi: 10.1016/j.mejo.2017.11.006.
- [BHJ10] D. Borodin and B. H. Juurlink, "Protective redundancy overhead reduction using instruction vulnerability factor," in Proc. ACM Int. Conf. Comput. Frontiers, 2010, pp. 319–326, doi: 10.1145/1787275.1787342.
- [BLG<sup>+</sup>20] A. Balakrishnan, T. Lange, M. Glorieux, D. Alexandrescu and M. Jenihhin, "Composing Graph Theory and Deep Neural Networks to Evaluate SEU Type Soft Error Effects," 2020 9th Mediterranean Conference on Embedded Computing (MECO), 2020, pp. 1-5, doi: 10.1109/MECO49872.2020.9134279.
- [BMS07] C. Bolchini, A. Miele and M. D. Santambrogio, "TMR and Partial Dynamic Reconfiguration to mitigate SEU faults in FPGAs," 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007), 2007, pp. 87-95, doi: 10.1109/DFT.2007.25.
- [BMX08] S. Bourdarie and M. Xapsos, "The Near-Earth Space Radiation Environment," in IEEE Transactions on Nuclear Science, vol. 55, no. 4, pp. 1810-1832, Aug. 2008, doi: 10.1109/TNS.2008.2001409.
- [BNW<sup>+</sup>05] B. Gill, M. Nicolaidis, F. Wolff, C. Papachristou and S. Garverick, "An efficient BICS design for SEUs detection and correction in semiconductor memories," Design, Automation and Test in Europe, 2005, pp. 592-597 Vol. 1, doi: 10.1109/DATE.2005.54.

- [BRC<sup>+</sup>05] A. Biswas, P. Racunas, R. Cheveresan, J. Emer, S. S. Mukherjee and R. Rangan, "Computing architectural vulnerability factors for address-based structures," 32nd International Symposium on Computer Architecture (ISCA'05), 2005, pp. 532-543, doi: 10.1109/ISCA.2005.18.
- [BSP<sup>+</sup>09] S. U. Bhandari, S. Subbaraman, S. Pujari, and R. Mahajan. 2009, "Real time video processing on FPGA using on the fly partial reconfiguration," In Proceedings of International Conference on Signal Processing Systems (ICSPPS). 244–247, doi: 10.1109/TNS.2021.3076400.
- [CAS<sup>+</sup>19] J. Chen, M. Andjelkovic, A. Simevski, Y. Li, P. Skoncej and M. Krstic, "Design of SRAM-Based Low-Cost SEU Monitor for Self-Adaptive Multiprocessing Systems," 2019 22nd Euromicro Conference on Digital System Design (DSD), 2019, pp. 514-521, doi: 10.1109/DSD.2019.00080.
- [CEN17] Community Coordinated Modeling Center. Coronal Mass Ejection Scoreboard, August 2017. URL: <https://kauai.ccmc.gsfc.nasa.gov/CMEscoreboard/>.
- [CIU<sup>+</sup>94] C. I. Underwood et al., "Radiation environment measurements with the cosmic ray experiments on-board the KITSAT-1 and PoSAT-1 micro-satellites," in IEEE Transactions on Nuclear Science, vol. 41, no. 6, pp. 2353-2360, Dec. 1994, doi: 10.1109/23.340587.
- [CJD<sup>+</sup>08] Copy J. Delorme et al., "A FPGA partial reconfiguration design approach for cognitive radio based on NoC architecture," 2008 Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference, 2008, pp. 355-358, doi: 10.1109/NEWCAS.2008.4606394.
- [CMA09] Clive Maxfield, editor, "FPGAs: World Class Designs," Newnes, Amsterdam; Boston, 1 edition, March 2009.
- [CRF22] J. A. Clemente, M. Rezaei and F. J. Franco, "Reliability of Error Correction Codes Against Multiple Events by Accumulation," in IEEE Transactions on Nuclear Science, vol. 69, no. 2, pp. 169-180, Feb. 2022, doi: 10.1109/TNS.2022.3143652.
- [CRE<sup>+</sup>19] D. Can, B. Richard, C. Estefanía, F. Maria Teresa, M. Arben, M. Andreas, U. Gerald Anton, G. Firat, "Disposable Sensors in Diagnostics, Food, and Environmental Monitoring". *Advanced Materials*. 31 (30): 1806739, 2019, doi:10.1002/adma.201806739.
- [CRE96] CREME96 tool, URL: <https://creme.isde.vanderbilt.edu/>.
- [CRL<sup>+</sup>06] A. K. Coskun, T. S. Rosing, Y. Leblebici, and G. De Micheli, "A simulation methodology for reliability analysis in multi-core SoCs," in Proc. ACM Great Lakes Symp. VLSI, 2006, pp. 95–99, doi: 10.1145/1127908.1127933.
- [CSB<sup>+</sup>15] A. I. Chumakov, A. V. Sogoyan, A. B. Boruzdina, A. A. Smolin and A. A. Pechenkin, "Multiple Cell Upset Mechanisms in SRAMs," 2015 15th European



- Conference on Radiation and Its Effects on Components and Systems (RADECS), 2015, pp. 1-5, doi: 10.1109/RADECS.2015.7365638.
- [CSH07] C. Claus, W. Stechele, and A. Herkersdorf, "Autovision—A run-time reconfigurable MPSoC architecture for future driver assistance systems, " *Information Technology* 49 (2007), doi: 10.1524/itit.2007.49.3.181.
- [CZM<sup>+</sup>07] C. Claus, J. Zeppenfeld, F. Muller and W. Stechele, "Using Partial-Run-Time Reconfigurable Hardware to accelerate Video Processing in Driver Assistance System," 2007 Design, Automation & Test in Europe Conference & Exhibition, 2007, pp. 1-6, doi: 10.1109/DATE.2007.364642.
- [DBG<sup>+</sup>15] L. Dilillo, A. Bosser, V. Gupta, F. Wrobel and F. Saigné, "Real-time SRAM based particle detector," 2015 6th International Workshop on Advances in Sensors and Interfaces (IWASI), 2015, pp. 58-62, doi: 10.1109/IWASI.2015.7184968.
- [DBP<sup>+</sup>14] J.M. Dutertre, R. Possamai Bastos, O. Potin, M.L. Flottes, B. Rouzeyre, G. Di Natale, A. Sarafianos, "Improving the ability of Bulk Built-In Current Sensors to detect Single Event Effects by using triple-well CMOS," *Microelectronics Reliability*, Volume 54, Issues 9–10, 2014, Pages 2289-2294, ISSN 0026-2714, doi: 10.1016/j.microrel.2014.07.151.
- [DLH<sup>+</sup>07] D. L. Hansen et al., "Correlation of Prediction to On-Orbit SEU Performance for a Commercial 0.25-um CMOS SRAM," in *IEEE Transactions on Nuclear Science*, vol. 54, no. 6, Dec. 2007, doi: 10.1109/TNS.2007.908787.
- [DNL08] D. C. Ness and D. J. Lilja, "Guiding Circuit Level Fault-Tolerance Design with Statistical Methods," 2008 Design, Automation and Test in Europe, 2008, pp. 348-353, doi: 10.1109/DATE.2008.4484704.
- [DPM<sup>+</sup>07] J. Delahaye, J. Palicot, C. Moy and P. Leray, "Partial Reconfiguration of FPGAs for Dynamical Reconfiguration of a Software Radio Platform," 2007 16th IST Mobile and Wireless Communications Summit, 2007, pp. 1-5, doi: 10.1109/ISTMWC.2007.4299250.
- [DSCO] Deep Space Climate Observatory (DSCOVR) satellite. URL:  
<https://www.nesdis.noaa.gov/current-satellite-missions/currently-flying/dscovr-deep-space-climate-observatory/>.
- [DWM03] P. E. Dodd, L. W. Massengill, "Basic mechanisms and modeling of single event upsets in digital microelectronics," *IEEE Transactions on Nuclear Science*, vol., 50, no. 3, 2003, doi: 10.1109/TNS.2003.813129.
- [ECH<sup>+</sup>16] E. Cheng et al., "CLEAR: Cross-layer exploration for architecting resilience: Combining hardware and software techniques to tolerate soft errors in processor cores," 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2016, pp. 1-6, doi: 10.1145/2897937.2897996.

- [ECA<sup>+</sup>19] E. Camporeale, "The challenge of machine learning in space weather: nowcasting and forecasting," *Space Weather*, 2019, doi: 10.1029/2018SW002061.
- [EDU13] Elena Dubrova, "Fault-Tolerant Design," Springer, New York, NY, March 2013.
- [EEZ17] Mark Engebretson, Eftyhia Zesta, "The Future of Ground Magnetometer Arrays in Support of Space Weather Monitoring and Research," , in *Space Weather*, 2017, doi: 10.1002/2017SW001718.
- [EHA07] E. Hannah, "Cosmic Ray Detectors for Integrated Circuit Chips," US Patent US7309866B2, 2007.
- [EPE11] E. Petersen, "Single Event Effects in Aerospace," Wiley-IEEE Press, 2011.
- [ERR85] Donald C. Ellison, Reuven Ramaty, " Shock acceleration of electrons and ions in solar flares," in *Astrophysical Journal*, Part 1 (ISSN 0004-637X), vol. 298, Nov. 1, 1985, p. 400-408. NASA-supported research, doi: 10.1086/163623.
- [ESS22] Euroconsult: Space and Satellite Sector Expert, 2022. URL: <https://www.euroconsult-ec.com/>.
- [FCB<sup>+</sup>19] Q. Fettes, M. Clark, R. Bunescu, A. Karanth and A. Louri, "Dynamic Voltage and Frequency Scaling in NoCs with Supervised and Reinforcement Learning Techniques," in *IEEE Transactions on Computers*, vol. 68, no. 3, pp. 375-389, 1 March 2019, doi: 10.1109/TC.2018.2875476.
- [FCH<sup>+</sup>15] F. Chollet, et al., Keras, 2015. URL: <https://keras.io>.
- [FDT<sup>+</sup>13] F. Wrobel, L. Dilillo, A. D. Touboul and F. Saigné, "Comparison of the transient current shapes obtained with the diffusion model and the double exponential law — Impact on the SER," 2013 14th European Conference on Radiation and Its Effects on Components and Systems (RADECS), 2013, pp. 1-4, doi: 10.1109/RADECS.2013.6937441.
- [FJN19] R. S. Ferreira and J. Nolte, "Low latency reconfiguration mechanism for fine-grained processor internal functional units," 2019 IEEE Latin American Test Symposium (LATS), 2019, pp. 1-6, doi: 10.1109/LATW.2019.8704560.
- [FNV<sup>+</sup>20] R. S. Ferreira, J. Nolte, F. Vargas, N. George and M. Hübner, "Run-time Hardware Reconfiguration of Functional Units to Support Mixed-Critical Applications," 2020 IEEE Latin-American Test Symposium (LATS), 2020, pp. 1-6, doi: 10.1109/LATS49555.2020.9093692.
- [FPE<sup>+</sup>11] F. Pedregosa, et al., "Scikit-learn: machine learning in Python," *J. Mach. Learn. Res.* 12 (2011) 2825–2830.
- [FRR<sup>+</sup>19] F. Rocha de Rosa, et al., "Using machine learning techniques to evaluate multicore soft error reliability," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2019, doi: 10.1109/TCSI.2019.2906155.

- [FSW<sup>+</sup>11] Farshad Firouzi, Mostafa E. Salehi, Fan Wang, Sied Mehdi Fakhraie, "An accurate model for soft error rate estimation considering dynamic voltage and frequency scaling effects," *Microelectronics Reliability*, *Microelectronics Reliability*, Volume 51, Issue 2, 2011, Pages 460-467, ISSN 0026-2714, doi:10.1016/j.microrel.2010.08.016.
- [FUK12] N. Farazmand, R. Ubal, and D. Kaeli, "Statistical fault injection- based AVF analysis of a GPU architecture," in *Proc. 8th IEEE Workshop Silicon Errors Logic Syst. Effects*, 2012.
- [GAU19] Geron, Aurelien, "Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems," 2nd ed., O'Reilly, 2019.
- [GFS<sup>+</sup>10] R. Ge, X. Feng, S. Song, H. Chang, D. Li and K. W. Cameron, "PowerPack: Energy Profiling and Analysis of High-Performance Systems and Applications," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 658-671, May 2010, doi: 10.1109/TPDS.2009.76.
- [GMR<sup>+</sup>17] R. Glein, P. Mengers, F. Rittner, R. Wansch and A. Heuberger, "BRAM implementation of a single-event upset sensor for adaptive single-event effect mitigation in reconfigurable FPGAs," 2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2017, pp. 1-8, doi: 10.1109/AHS.2017.8046352
- [GOES] Geostationary Operational Environmental Satellites (GOES) - space environment monitor (SEM) database. URL: <https://ngdc.noaa.gov/stp/satellite/goes/dataaccess.html>.
- [GRH16] R. Glein, F. Rittner and A. Heuberger, "Detection of solar particle events inside FPGAs," 2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS), 2016, pp. 1-5, doi: 10.1109/RADECS.2016.8093159.
- [GRH18] R. Glein, F. Rittner and A. Heuberger, "Adaptive single-event effect mitigation for dependable processing systems based on FPGAs," *Microprocessors and Microsystems*, volume 59, 2018, doi: 10.1016/j.micpro.2018.03.004.
- [GSZ09] B. Gill, N. Seifert and V. Zia, "Comparison of alpha particle and neutron-induced combinational and sequential logic error rates at the 32nm technology node," 2009 IEEE International Reliability Physics Symposium, 2009, pp. 199-205, doi: 10.1109/IRPS.2009.5173251.
- [GTS<sup>+</sup>14] G. Tsiligiannis et al., "An SRAM Based Monitor for Mixed-Field Radiation Environments," in *IEEE Transactions on Nuclear Science*, vol. 61, no. 4, pp. 1663-1670, Aug. 2014, doi: 10.1109/TNS.2014.2299733.

- [GWF07] G. Wirth and C. Fayomi, "The Bulk Built In Current Sensor Approach for Single Event Transient Detection," 2007 International Symposium on System-on-Chip, 2007, pp. 1-4, doi: 10.1109/ISSOC.2007.4427422.
- [HBH<sup>+</sup>18] M. Havranek, T. Benka, M. Hejtmanek, Z. Janoska, V. Kafka, J. Kopecek, M. Kuklova, M. Marcisovska, M. Marcisovsky, G. Neue, P. Svihra, L. Tomasek, P. Vancura, V. Vrba, "MAPS Sensor for Radiation Imaging Designed in 180 nm SOI CMOS Technology," *Journal of Instrumentation*, Volume 13, Number 06, Pages C06004--C06004, 2018, doi: 10.1088/1748-0221/13/06/c06004.
- [HBS14] H. Hussain, K. Benkrid, and H. Seker. 2014, "Novel dynamic partial reconfiguration implementations of the support vector machine classifier on FPGA," *Turkish Journal of Electrical Engineering & Computer Sciences* 24, 2014, doi: 10.1109/TNS.2021.3076400.
- [HED04] S. E. Hoyos, H. D. R. Evans and E. Daly, "From Satellite ion flux data to SEU rate estimation," in *IEEE Transactions on Nuclear Science*, vol. 51, no. 5, pp. 2927-2935, Oct. 2004, doi: 10.1109/TNS.2004.835072.
- [HGR<sup>+</sup>05] R. Harboe-Sorensen, F. -. Guerre and A. Roseng, "Design, Testing and Calibration of a "Reference SEU Monitor" System," 2005 8th Conference on Radiation and Its Effects on Components and Systems, 2005, pp. B3-1-B3-7, doi: 10.1109/RADECS.2005.4365561.
- [HKE12] H. M. Hussain, K. Benkrid, A. Ebrahim, A. T. Erdogan, and H. Seker, "Novel dynamic partial reconfiguration implementation of K-means clustering on FPGAs: Comparative results with GPPs and GPUs," *International Journal of Reconfigurable Computing (IJRC)*, 2012, doi: 10.1155/2012/135926.
- [HMB<sup>+</sup>18] H.M. Bain, et al., "Using machine learning techniques to forecast solar energetic particles," *Proc. SOARS*, 2018, doi: 10.5065/kzsz-vf38.
- [HSW<sup>+</sup>09] J. Heiner, B. Sellers, M. Wirthlin and J. Kalb, "FPGA partial reconfiguration via configuration scrubbing," 2009 International Conference on Field Programmable Logic and Applications, 2009, pp. 99-104, doi: 10.1109/FPL.2009.5272543.
- [HSW00] P. Hazucha, C. Svensson and S. A. Wender, "Cosmic-ray soft error rate characterization of a standard 0.6-/spl mu/m CMOS process," in *IEEE Journal of Solid-State Circuits*, vol. 35, no. 10, pp. 1422-1429, Oct. 2000, doi: 10.1109/4.871318.
- [HUS17] Hu, S., "Solar particle events and radiation exposure in space," *NASA Space Radiation Program Element, Human Research Program*, pp. 1–15, 2017.
- [ICR12] International Commission on Radiation Units and Measurements, "Fundamental Quantities and Units for Ionizing Radiation," *Radiation Protection Dosimetry*, Volume 150, Issue 4, July 2012, doi:10.1093/rpd/ncs077.

- [IEC05] Functional Safety of electrical / electronic / programmable electronic safety related systems (IEC 61508), International Electrotechnical Commission, 2005.
- [IFF04] F. Irom and F. F. Farmanesh, "Frequency dependence of single-event upset in advanced commercial PowerPC microprocessors," in *IEEE Transactions on Nuclear Science*, vol. 51, no. 6, pp. 3505-3509, Dec. 2004, doi: 10.1109/TNS.2004.839111.
- [JBE<sup>+</sup>12] J. Bergstra, et al., "Random search for hyper-parameter optimization, " *J. Mach. Learn. Res.* 13 (2012) 281–305 Feb. URL: <https://www.jmlr.org/papers/v13/bergstra12a.html>.
- [JCG<sup>+</sup>12] Adam Jacobs, Grzegorz Cieslewski, Alan D. George, Ann Gordon-Ross, Herman Lam, "Reconfigurable Fault Tolerance: A Comprehensive Framework for Reliable and Adaptive FPGA-Based Space Computing," *ACM Transactions on Reconfigurable Technology and Systems*, volume 5, issue 4, 2012, doi: 10.1145/2392616.2392619.
- [JCK18] J. Chen, M. Krstic, "Engineering of Cross-Layer Fault Tolerance In Multiprocessing Systems," *Proc. PhD Forum of the 8th BELAS Summer School*, Tallinn, June 20 - 22, 2018, Estonia
- [JCP80] J.C. Pickel, et al., "Cosmic-ray-induced errors in MOS devices," *IEEE Trans. Nucl. Sci.* 27 (2) (1980) 1006–1015. April, doi: 10.1109/TNS.1980.4330967.
- [JGW08] J. G. Wirth, "Bulk Built-In Current Sensors for Single Event Transient Detection in Deep-Submicron Technologies," *Microelectronics Reliability*, Volume 48, Issue 5, Pages 710-715, ISSN 0026-2714, 2008, doi:10.1016/j.microrel.2008.01.002.
- [JHO<sup>+</sup>11] J. Howard, et al., "A 48-Core IA-32 Processor in 45 nm CMOS Using On-Die Message-Passing and DVFS for Performance and Power Scaling," in *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 173-183, Jan. 2011, doi: 10.1109/JSSC.2010.2079450.
- [JMD<sup>+</sup>14] J. M. Dutertre et al., "Improving the Ability of Bulk Built-In Current Sensors to Detect Single Event Effects by Using Triple-Well CMOS," *Microelectronics Reliability*, 2014, doi: 10.1016/j.microrel.2014.07.151.
- [JNK07] J. Noguera and I. O. Kennedy, "Power Reduction in Network Equipment Through Adaptive Partial Reconfiguration," *2007 International Conference on Field Programmable Logic and Applications*, 2007, pp. 240-245, doi: 10.1109/FPL.2007.4380654.
- [JWW10] Tylka. Allan J., Dietrich. William, Atwell. William, "Band Function Representations of Solar Proton Spectra in Ground-Level Events," *38th COSPAR Scientific Assembly*. Held 18-15 July 2010, in Bremen, Germany, p.4, 2010.

- [KHY<sup>+</sup>14] K. H. Yearby, et al., "Single-event upsets in the cluster and double star digital wave processor instruments," *Space Weather*, vol. 12, no. 1, Jan. 2014, pp. 24-28, doi: 10.1002/2013SW000985.
- [KMO<sup>+</sup>09] S. Kasahara, T. Mitani, K. Ogasawara, T. Takashima, M. Hirahara, K. Asamura, "Application of single-sided silicon strip detector to energy and charge state measurements of medium energy ions in space," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Volume 603, Issue 3, 2009, Pages 355-360, ISSN 0168-9002, doi: 10.1016/j.nima.2009.02.004.
- [KRL<sup>+</sup>93] R. Koga, S. D. Pinkerton, T. J. Lie and K. B. Crawford, "Single-word multiple-bit upsets in static random access devices," in *IEEE Transactions on Nuclear Science*, vol. 40, no. 6, pp. 1941-1946, Dec. 1993, doi: 10.1109/23.273460.
- [KRS<sup>+</sup>15] Kristian S. et al., "Design and characterization of an SRAM-based neutron detector for particle therapy," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Volume 804, 2015, Pages 64-71, doi: 10.1016/j.nima.2015.09.049.
- [KVF19] Kizheppatt Vipin, Suhaib A. Fahmy, "FPGA Dynamic and Partial Reconfiguration: A Survey of Architectures, Methods, and Applications," in *ACM computing Surveys*, volume 51, issue 4, 2019, doi: 10.1145/3193827.
- [LBA<sup>+</sup>20] Y. Li, A. Breitenreiter, M. Andjelkovic, J. Chen, M. Babic, M. Krstic., "Double cell upsets mitigation through triple modular redundancy," *Microelectronics Journal* Volume 96, 2020, doi: 10.1016/j.mejo.2019.104683.
- [LCB<sup>+</sup>21] L. Lu, J. Chen, A. Breitenreiter, O. Schrape, M. Ulbricht and M. Krstic, "Machine Learning Approach for Accelerating Simulation-based Fault Injection," 2021 IEEE Nordic Circuits and Systems Conference (NorCAS), 2021, pp. 1-6, doi: 10.1109/NorCAS53631.2021.9599646.
- [LRX<sup>+</sup>19] J. Li, P. Reviriego, L. Xiao and A. Klockmann, "Protecting Large Word Size Memories against MCUs with 3-bit Burst Error Correction," 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2019, pp. 1-4, doi: 10.1109/DFT.2019.8875396.
- [MAN<sup>+</sup>20] M. Andjelkovic et al., "Design of Radiation Hardened RADFET Readout System for Space Applications," 2020 23rd Euromicro Conference on Digital System Design (DSD), 2020, pp. 484-488, doi: 10.1109/DSD51259.2020.00082.
- [MAN22] M. Andjelkovic, "A Methodology for Characterization, Modeling and Mitigation of Single Event Transient Effects in CMOS Standard Combinational Cells," PhD dissertation, University of Potsdam, 2022.

- [MAX<sup>+</sup>00] M. A. Xapsos et al., "Characterizing solar proton energy spectra for radiation effects applications," in *IEEE Transactions on Nuclear Science*, vol. 47, no. 6, pp. 2218-2223, Dec. 2000, doi: 10.1109/23.903756.
- [MBA<sup>+</sup>20] M. Bagatin et al., "A Heavy-Ion Detector Based on 3-D NAND Flash Memories," in *IEEE Transactions on Nuclear Science*, vol. 67, no. 1, pp. 154-160, Jan. 2020, doi: 10.1109/TNS.2019.2955776.
- [MBR<sup>+</sup>16] M. Brucoli et al., "A complete qualification of floating gate dosimeter for CERN applications," 2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS), 2016, pp. 1-4, doi: 10.1109/RADECS.2016.8093162.
- [MCA<sup>+</sup>09] M. Caffrey et al., "On-Orbit Flight Results from the Reconfigurable Cibola Flight Experiment Satellite (CFESat)," 2009 17th IEEE Symposium on Field Programmable Custom Computing Machines, 2009, pp. 3-10, doi: 10.1109/FCCM.2009.22.
- [MCE<sup>+</sup>03] M. Ceschia et al., "Identification and classification of single-event upsets in the configuration memory of SRAM-based FPGAs," in *IEEE Transactions on Nuclear Science*, vol. 50, no. 6, pp. 2088-2094, Dec. 2003, doi: 10.1109/TNS.2003.821411.
- [MGA<sup>+</sup>17] M. Gautschi et al., "Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2700-2713, Oct. 2017, doi: 10.1109/TVLSI.2017.2654506.
- [MGL14] Max Gleber, "CME Week: The Difference Between Flares and CMEs," Sep. 2014, URL: <https://www.nasa.gov/content/goddard/the-difference-between-flares-and-cmes/>.
- [MHB13] Matthew Marinella and Hugh Barnaby. "Total ionizing dose and displacement damage effects in embedded memory technologies (tutorial notes-draft 1)," Technical report, Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), 2013.
- [MIL15] F. M. M. u. Islam and M. Lin, "A Framework for Learning Based DVFS Technique Selection and Frequency Scaling for Multi-core Real-Time Systems," 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, 2015, pp. 721-726, doi: 10.1109/HPCC-CSS-ICSS.2015.313.
- [MKR<sup>+</sup>21] M. Krstic et al., "Cross-Layer Digital Design Flow for Space Applications," 2021 IEEE 32nd International Conference on Microelectronics (MIEL), 2021, pp. 45-54, doi: 10.1109/MIEL52794.2021.9569065.

- [MSC<sup>+</sup>17] S. K. Morley, J. P. Sullivan, M. R. Carver, R. M. Kippen, R. H. W. Friedel, G. D. Reeves, M. G. Henderson, "Energetic Particle Data From the Global Positioning System Constellation," in *Space Weather*, 2017, doi: 10.1002/2017SW001604.
- [MSC17] Marie Skłodowska-Curie RESCUE action, 2017. URL: <http://www.rescue-etn.eu/>.
- [MSC10] M. Schölzel, "HW/SW co-detection of transient and permanent faults with fast recovery in statically scheduled data paths," 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), 2010, pp. 723-728, doi: 10.1109/DATE.2010.5456957.
- [MTR<sup>+</sup>84] R.E. McGuire, T.T. von Rosenvinge, "The energy spectra of solar energetic particles," *Advances in Space Research*, Volume 4, Issues 2–3, 1984, doi: 10.1016/0273-1177(84)90301-6.
- [MWE<sup>+</sup>03] S. S. Mukherjee, C. Weaver, J. Emer, S. K. Reinhardt and T. Austin, "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor," *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture*, 2003. MICRO-36., 2003, pp. 29-40, doi: 10.1109/MICRO.2003.1253181.
- [MYH70] M. Y. Hsiao, "A Class of Optimal Minimum Odd-weight-column SEC-DED Codes," in *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 395-401, July 1970, doi: 10.1147/rd.144.0395.
- [NAD<sup>+</sup>14] N. A. Dodds et al., "Hardness Assurance for Proton Direct Ionization-Induced SEEs Using a High-Energy Proton Beam," in *IEEE Transactions on Nuclear Science*, vol. 61, no. 6, pp. 2904-2914, Dec. 2014, doi: 10.1109/TNS.2014.2364953.
- [NRV<sup>+</sup>05] E. H. Neto, I. Ribeiro, M. Vieira, G. Wirth and F. L. Kastensmidt, "Evaluating Fault Coverage of Bulk Built-in Current Sensor for Soft Errors in Combinational and Sequential Logic," 2005 18th Symposium on Integrated Circuits and Systems Design, 2005, pp. 62-67, doi: 10.1109/SBCCI.2005.4286833.
- [NOAA] National Oceanic and Atmospheric Administration (NOAA) solar proton events affecting the earth environment lists. URL: <https://www.ngdc.noaa.gov/stp/satellite/goes/doc/SPE.txt>.
- [NYS<sup>+</sup>05] H. T. Nguyen, Y. Yagil, N. Seifert and M. Reitsma, "Chip-level soft error estimation method," in *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 365-381, Sept. 2005, doi: 10.1109/TDMR.2005.858334.
- [PIM19] Pignol M., "System Hardening and Real Applications," In: Velazco R., McMorro D., Estela J. (eds) *Radiation Effects on Integrated Circuits and Systems for Space Applications*, 2019. Springer, Cham, doi: 10.1007/978-3-030-04660-6\_9.
- [PKS17] R. Psiakis, A. Kritikakou and O. Sentieys, "Run-time Instruction Replication for permanent and soft error mitigation in VLIW processors," 2017 15th IEEE



- International New Circuits and Systems Conference (NEWCAS), 2017, pp. 321-324, doi: 10.1109/NEWCAS.2017.8010170.
- [PPA<sup>+</sup>92] E. L. Petersen, J. C. Pickel, J. H. Adams and E. C. Smith, "Rate prediction for single event effects-a critique," in *IEEE Transactions on Nuclear Science*, vol. 39, no. 6, pp. 1577-1599, Dec. 1992, doi: 10.1109/23.211340.
- [PPS<sup>+</sup>93] E. L. Petersen, J. C. Pickel, E. C. Smith, P. J. Rudeck and J. R. Letaw, "Geometrical factors in SEE rate calculations," in *IEEE Transactions on Nuclear Science*, vol. 40, no. 6, pp. 1888-1909, Dec. 1993, doi: 10.1109/23.273465.
- [PRG<sup>+</sup>16] E. Pikhay, Y. Roizin, U. Gatti and C. Calligaro, "Re-usable 180nm CMOS dosimeter based on a floating gate device," 2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2016, pp. 125-128, doi: 10.1109/ICECS.2016.7841148.
- [PTB<sup>+</sup>19] J. Prinzie, S. Thys, B. Van Bockel, J. Wang, V. De Smedt and P. Leroux, "An SRAM-Based Radiation Monitor With Dynamic Voltage Control in 0.18 um CMOS Technology," in *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 282-289, Jan. 2019, doi: 10.1109/TNS.2018.2885693.
- [PULP] PULPissimo open-source microcontroller architecture. URL: <https://github.com/pulp-platform/pulpissimo>.
- [QCU<sup>+</sup>19] Q. Cui et al., "Stochastic Online Learning for Mobile Edge Computing: Learning from Changes," in *IEEE Communications Magazine*, vol. 57, no. 3, pp. 63-69, March 2019, doi: 10.1109/MCOM.2019.1800644.
- [RCC<sup>+</sup>19] S. Roy, S. Chakraborty, S. Chatterjee, S. Biswas, S. Das, S.K. Ghosh, A. Maulik, S. Raha, "Plastic scintillator detector array for detection of cosmic ray air shower," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Volume 936, 2019, Pages 249-251, ISSN 0168-9002, doi: 10.1016/j.nima.2018.09.109.
- [RDS<sup>+</sup>08] R. D. Schrimpf et al., "Reliability and radiation effects in IC technologies," 2008 IEEE International Reliability Physics Symposium, 2008, pp. 97-106, doi: 10.1109/RELPHY.2008.4558869.
- [RHM<sup>+</sup>17] A. M. Rahmani, M. Haghbayan, A. Miele, P. Liljeberg, A. Jantsch and H. Tenhunen, "Reliability-Aware Runtime Power Management for Many-Core Systems in the Dark Silicon Era," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 2, pp. 427-440, Feb. 2017, doi: 10.1109/TVLSI.2016.2591798.
- [RHS<sup>+</sup>01] R. Harboe-Sørensen, et al., "Observation and analysis of Single Event Effects on-board the SOHO satellite," 2001 6th European Conference on Radiation and Its Effects on Components and Systems, doi: 10.1109/RADECS.2001.1159256.

- [RKL10] R. Khraisha and J. Lee, "A scalable H.264/AVC deblocking filter architecture using dynamic partial reconfiguration," 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 1566-1569, doi: 10.1109/ICASSP.2010.5495525.
- [RLR<sup>+</sup>11] A. Rahimi, I. Loi, M. R. Kakoei and L. Benini, "A fully-synthesizable single-cycle interconnection network for Shared-L1 processor clusters," 2011 Design, Automation & Test in Europe, 2011, pp. 1-6, doi: 10.1109/DATE.2011.5763085.
- [RPB<sup>+</sup>20] R. Possamai Bastos et al., "Assessment of On-Chip Current Sensor for Detection of Thermal-Neutron-Induced Transients," in IEEE Transactions on Nuclear Science, vol. 67, no. 7, pp. 1404-1411, July 2020, doi: 10.1109/TNS.2020.2975923.
- [RPN<sup>+</sup>20] N. M. Rezk, M. Purnaprajna, T. Nordström and Z. UIAbdin, "Recurrent Neural Networks: An Embedded Computing Perspective," in IEEE Access, vol. 8, pp. 57967-57996, 2020, doi: 10.1109/ACCESS.2020.2982416.
- [RSA<sup>+</sup>21] R. Sarić et al., "Classification of Space Particle Events using Supervised Machine Learning Algorithms," 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), 2021, pp. 1-10, doi: 10.1109/DSAA53316.2021.9564114.
- [RSE<sup>+</sup>16] R. Secondo et al., "Embedded Detection and Correction of SEU Bursts in SRAM Memories Used as Radiation Detectors," in IEEE Transactions on Nuclear Science, vol. 63, no. 4, pp. 2168-2175, Aug. 2016, doi: 10.1109/TNS.2016.2521485.
- [RTU00] R. Turner, "Solar particle events from a risk management perspective," in IEEE Transactions on Plasma Science, vol. 28, no. 6, pp. 2103-2113, Dec. 2000, doi: 10.1109/27.902237.
- [RWH50] R. W. Hamming, "Error detecting and error correcting codes," in The Bell System Technical Journal, vol. 29, no. 2, pp. 147-160, April 1950, doi: 10.1002/j.1538-7305.1950.tb00463.x.
- [SBS17] E. G. Stassinopoulos, J. L. Barth and C. A. Stauffer, "Measurement of Cosmic Ray and Trapped Proton LET Spectra on the STS-95 HOST Mission," in IEEE Transactions on Nuclear Science, vol. 64, no. 8, pp. 2007-2015, Aug. 2017, doi: 10.1109/TNS.2017.2665439.
- [SHI<sup>+</sup>16] S. Hirokawa, et al., "Multiple sensitive volume based soft error rate estimation with machine learning," Proc. European Conference on Radiation and its Effects on Components and Systems (RADECS), 2016, doi: 10.1109/RADECS.2016.8093181.
- [SHK<sup>+</sup>12] A. Simevski, E. Hadzieva, R. Kraemer and M. Krstic, "Scalable design of a programmable NMR voter with inputs' state descriptor and self-checking capability," 2012 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2012, pp. 182-189, doi: 10.1109/AHS.2012.6268648.

- [SJP<sup>+</sup>08] E. Seo, J. Jeong, S. Park, and J. Lee, "Energy efficient scheduling of real-time tasks on multicore processors," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 11, pp. 1540–1552, Nov 2008, doi: 10.1109/TPDS.2008.104.
- [SKA<sup>+</sup>09] S. Kasahara et al., "Application on Single-Sided Silicon Strip Detectors for Energy and Charge State Measurements of Medium Energy Ions in Space," *Nuclear Instruments and Methods in Physics Research A*, 2009, doi: 10.1016/j.nima.2009.02.004.
- [SKH11] Stefan K. Hoffgen, "Dosis und Struktureffekte," Fraunhofer Institute for Technological Trend Analysis INT, Nov. 2011.
- [SKK14] A. Simevski, R. Kraemer and M. Krstic, "Increasing multiprocessor lifetime by Youngest-First Round-Robin core gating patterns," 2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2014, pp. 233-239, doi: 10.1109/AHS.2014.6880182.
- [SKK11] A. Simevski, R. Kraemer and M. Krstic, "Low-complexity integrated circuit aging monitor," 14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems, 2011, pp. 121-125, doi: 10.1109/DDECS.2011.5783060.
- [SKK<sup>+</sup>14] S. Srinivasan, N. Kurella, I. Koren, S. Kundu and R. Rodrigues, "A runtime support mechanism for fast mode switching of a self-morphing core for power efficiency," 2014 23rd International Conference on Parallel Architecture and Compilation Techniques (PACT), 2014, pp. 491-492, doi: 10.1145/2628071.2628124.
- [SKK15] S. Srinivasan, I. Koren and S. Kundu, "Online mechanism for reliability and power-efficiency management of a dynamically reconfigurable core," 2015 33rd IEEE International Conference on Computer Design (ICCD), 2015, pp. 327-334, doi: 10.1109/ICCD.2015.7357121.
- [SLJ12] X. She, N. Li and D. W. Jensen, "SEU Tolerant Memory Using Error Correction Code," in *IEEE Transactions on Nuclear Science*, vol. 59, no. 1, pp. 205-210, Feb. 2012, doi: 10.1109/TNS.2011.2176513.
- [SML<sup>+</sup>01] N. Seifert, D. Moyer, N. Leland and R. Hokinson, "Historical trend in alpha-particle induced soft error rates of the Alpha/sup TM/ microprocessor," 2001 IEEE International Reliability Physics Symposium Proceedings. 39th Annual (Cat. No.00CH37167), 2001, pp. 259-265, doi: 10.1109/RELPHY.2001.922911.
- [SOHO] Solar and Heliospheric Observatory spacecraft. URL: <https://sohowww.nascom.nasa.gov/>.
- [SPG<sup>+</sup>15] L. Saiz-Adalid, P. Reviriego, P. Gil, S. Pontarelli and J. A. Maestro, "MCU Tolerance in SRAMs Through Low-Redundancy Triple Adjacent Error Correction," in *IEEE Transactions on Very Large Scale Integration (VLSI)*

- Systems, vol. 23, no. 10, pp. 2332-2336, Oct. 2015, doi: 10.1109/TVLSI.2014.2357476.
- [SRO<sup>+</sup>19] S. Roy et al., "Plastic Scintillator Detector Array for Detection of Cosmic Ray Air Shower," Nuclear Instruments and Methods in Physics Research A, 2019, doi: 10.1016/j.nima.2018.09.109.
- [SRG<sup>+</sup>15] L. Saiz-Adalid, P. Reviriego, P. Gil, S. Pontarelli and J. A. Maestro, "MCU Tolerance in SRAMs Through Low-Redundancy Triple Adjacent Error Correction," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 10, pp. 2332-2336, Oct. 2015, doi: 10.1109/TVLSI.2014.2357476.
- [SRK09] V. Sridharan and D. R. Kaeli, "Eliminating microarchitectural dependency from Architectural Vulnerability," 2009 IEEE 15th International Symposium on High Performance Computer Architecture, 2009, pp. 117-128, doi: 10.1109/HPCA.2009.4798243.
- [SRK10] V. Sridharan and D. R. Kaeli, "Using hardware vulnerability factors to enhance AVF analysis," in Proc. Int. Symp. Comput. Archit., 2010, pp. 461-472, doi: 10.1145/1816038.1816023.
- [SSB<sup>+</sup>20] A. Simevski, O. Schrape, C. Benito, M. Krstic and M. Andjelkovic, "PISA: Power-robust Multiprocessor Design for Space Applications," 2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS), 2020, pp. 1-6, doi: 10.1109/IOLTS50870.2020.9159716.
- [SSB98] E.G Stassinopoulos, Craig A Stauffer, G.J Brucker, "Miniature high-let radiation spectrometer for space and avionics applications," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 416, Issues 2-3, 1998, Pages 531-535, ISSN 0168-9002, doi: 10.1016/S0168-9002(98)00633-0.
- [SSC<sup>+</sup>19] A. Simevski, P. Skoncej, C. Calligaro and M. Krstic, "Scalable and Configurable Multi-Chip SRAM in a Package for Space Applications," 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2019, pp. 1-6, doi: 10.1109/DFT.2019.8875489.
- [SSM00] P. P. Shirvani, N. R. Saxena and E. J. McCluskey, "Software-implemented EDAC protection against SEUs," in IEEE Transactions on Reliability, vol. 49, no. 3, pp. 273-284, Sept 2000, doi: 10.1109/24.914544.
- [STW11] Dimitrios Serpanos, Tilman Wolf, "Specialized hardware components," Architecture of Network Systems, Morgan Kaufmann, 2011, ISSN 15459888.
- [SVF<sup>+</sup>13] S. Shreejith, K. Vipin, S. A. Fahmy and M. Lukasiewicz, "An approach for redundancy in FlexRay networks using FPGA partial reconfiguration," 2013

- Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013, pp. 721-724, doi: 10.7873/DATE.2013.155.
- [SVK<sup>+</sup>16] S. S. Sahoo, B. Veeravalli and A. Kumar, "Cross-layer fault-tolerant design of real-time systems," 2016 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2016, pp. 63-68, doi: 10.1109/DFT.2016.7684071.
- [TDB<sup>+</sup>96] A. J. Tylka, W. F. Dietrich, P. R. Boberg, E. C. Smith and J. H. Adams, "Single event upsets caused by solar energetic heavy ions," in IEEE Transactions on Nuclear Science, vol. 43, no. 6, pp. 2758-2766, Dec. 1996, doi: 10.1109/23.556863.
- [TDB97] A. J. Tylka, W. F. Dietrich and P. R. Bobery, "Probability distributions of high-energy solar-heavy-ion fluxes from IMP-8: 1973-1996," in IEEE Transactions on Nuclear Science, vol. 44, no. 6, pp. 2140-2149, Dec. 1997, doi: 10.1109/23.659029.
- [TEC08] The European Cooperation for Space Standardization editor, "Space engineering: Methods for the calculation of radiation received and its effects, and a policy for design margins," ECSS-E-ST-10-12C, Nov. 2008.
- [TFZ08] Copy H. Taghipour, J. Frounchi and M. H. Zarifi, "Design and implementation of MP3 decoder using partial dynamic reconfiguration on Virtex-4 FPGAs," 2008 International Conference on Computer and Communication Engineering, 2008, pp. 683-686, doi: 10.1109/ICCCE.2008.4580691.
- [TKK<sup>+</sup>17] Q. Tang, S. Kumar, C. H. Kim and D. E. Fulkerson, "A compact high-sensitivity 2-transistor radiation sensor array," 2017 IEEE International Reliability Physics Symposium (IRPS), 2017, pp. SE-7.1-SE-7.4, doi: 10.1109/IRPS.2017.7936408.
- [TLA<sup>+</sup>19] T. Lange, et al, "Machine learning to tackle the challenges of transient and soft errors in complex circuits, " Proc. 25th IEEE International Symposium on Online Testing and Robust System Design, 2019, doi: 10.1109/IOLTS.2019.8854423.
- [TMM97] Tom M. Mitchell, "Machine Learning," Blacklick, Ohio, USA: McGraw-Hill Science/Engineering/Math, 1997; ISBN-13: 978-0070428072.
- [TSF08] Toshinori Sato and Toshimasa Funaki, "Dependability, power, and performance trade-off on a multicore processor," 2008 Asia and South Pacific Design Automation Conference, 2008, doi: 10.1109/ASPAC.2008.4484044.
- [UCS22] UCS Satellite Database, 2022. URL: <https://www.ucsusa.org/resources/satellite-database>.
- [UVG16] G. Upasani, X. Vera and A. González, "A Case for Acoustic Wave Detectors for Soft-Errors," in IEEE Transactions on Computers, vol. 65, no. 1, pp. 5-18, 1 Jan. 2016, doi: 10.1109/TC.2015.2419652.

- [VGU17] V. Gupta, "Analysis of Single Event Radiation Effects and Fault Mechanisms in SRAM, FRAM and NAND Flash: Application to the MTCube Nanosatellite Project," PhD dissertation, University of Montpellier, 2017.
- [VPK15] V. Petrovic and M. Krstic, "Design Flow for Radhard TMR Flip-Flops," 2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, 2015, pp. 203-208, doi: 10.1109/DDECS.2015.65.
- [VSZ<sup>+</sup>17] V. Sze, et al., "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," in *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295-2329, Dec. 2017, doi: 10.1109/JPROC.2017.2761740.
- [WAB<sup>+</sup>20] W.S. Wong, J. Alozy, R. Ballabriga, M. Campbell, I. Kremastiotis, X. Llopart, T. Poikela, V. Sriskaran, L. Tlustos, D. Turecek, "Introducing Timepix2, a frame-based pixel detector readout ASIC measuring energy deposition and arrival time," *Radiation Measurements*, Volume 131, 2020, 106230, ISSN 1350-4487, doi: 10.1016/j.radmeas.2019.106230.
- [WEW14] Weisstein, Eric W., "Radiation". Eric Weisstein's World of Physics, Wolfram Research, January 2014.s
- [WHG07] K. R. Walcott, G. Humphreys, and S. Gurusurthi, "Dynamic prediction of architectural vulnerability from microarchitectural state," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 516-527, 2007, doi: 10.1145/1250662.1250726.
- [WHL<sup>+</sup>15] Wang, H., Liu, R., Chen, L. et al. A Novel Built-in Current Sensor for N-WELL SET Detection. *J Electron Test* 31, 395-401 (2015), doi: 10.1007/s10836-015-5538-0.
- [WIND] Global Geospace Science (GGS) Wind satellite. URL: <https://wind.nasa.gov/>.
- [WSW<sup>+</sup>20] W. S. Wong et al, "Introducing Timepix2, A Frame-Based Pixel Detector Readout ASIC Measuring Energy Deposition and Arrival Time," *Radiation Measurements*, 2020, doi: 10.1016/j.radmeas.2019.106230.
- [YCC95] Xiao-Hu Yu, Guo-An Chen and Shi-Xin Cheng, "Dynamic learning rate optimization of the backpropagation algorithm," in *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 669-677, May 1995, doi: 10.1109/72.377972.
- [YVL<sup>+</sup>15] Kristian S. Ytre-Hauge, Arild Velure, Eivind F. Larsen, Camilla H. Stokkevåg, Dieter Röhrich, Njål Brekke, Odd Harald Odland, "Design and characterization of an SRAM-based neutron detector for particle therapy," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Volume 804, 2015, Pages 64-71, ISSN 0168-9002, doi: 10.1016/j.nima.2015.09.049.

- [YWL<sup>+</sup>08] S. Yang, W. Wang, T. Lu, W. Wolf, N. Vijaykrishnan and Y. Xie, "Case Study of Reliability-Aware and Low-Power Design," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 7, pp. 861-873, July 2008, doi: 10.1109/TVLSI.2008.2000460.
- [YWZ05] J. Yan and W. Zhang, "Compiler-guided register reliability improvement against soft errors," in *Proc. ACM Int. Conf. Embedded Softw.*, 2005, pp. 203–209, doi: 10.1145/1086228.1086266.
- [ZMM04] Dakai Zhu, R. Melhem and D. Mosse, "The effects of energy management on reliability in real-time embedded systems," *IEEE/ACM International Conference on Computer Aided Design*, 2004. ICCAD-2004., 2004, pp. 35-40, doi: 10.1109/ICCAD.2004.1382539.
- [ZRY<sup>+</sup>13] Zhang, Z., Ren, Y., Chen, L. et al. "A Bulk Built-In Voltage Sensor to Detect Physical Location of Single-Event Transients," *J Electron Test* 29, 249–253 (2013), doi: 10.1007/s10836-013-5364-1.
- [ZSW00] P. Hazucha, C. Svensson and S. A. Wender, "Cosmic-ray soft error rate characterization of a standard 0.6-/spl mu/m CMOS process," in *IEEE Journal of Solid-State Circuits*, vol. 35, no. 10, pp. 1422-1429, Oct. 2000, doi: 10.1109/4.871318.
- [ZZH<sup>+</sup>13] Z. Zhang et al., "A Bulk Built-In Voltage Sensor to Detect Physical Location of Single-Event Transients," *Journal of Electronic Testing*, 2013, doi: 10.1007/s10836-013-5364-1.