Monitoring of temperature effects on CMOS memories

Emna Farjallah

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Systèmes Automatiques et Microélectroniques

École doctorale : Information, Structures, Systèmes

Unité de recherche : Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier

Monitoring des Effets de la Température sur les Mémoires CMOS

Présentée par Emna FARJALLAH

Le 27 Novembre 2018

Sous la direction de Luigi DILILLO

Devant le jury composé de

Bruno ROUZEYRE, Professeur, Univ. de Montpellier	Président du jury
Alberto BOSIO, Professeur, Univ. de Lyon	Rapporteur
Eduardo Augusto BEZERRA, Professeur, Univ. de Santa Catarina	Rapporteur
Marco OTTAVI, Professeur, Univ. de Rome Tor Vergata	Examinateur
Frédéric WROBEL, Professeur, Univ. de Montpellier	Examinateur
Hassen AZIZA, Maître de conférences, École polytech' de Marseille	Examinateur
Valentin GHERMAN, Docteur chercheur, CEA LIST	Invité
Jean-Marc ARMANI, Ingénieur chercheur, CEA LIST	Invité

Monitoring des effets de la température sur les mémoires CMOS

Résumé: La complexité des systèmes électroniques ne cesse d'augmenter, tout comme la tendance actuelle de miniaturisation des transistors. La fiabilité est ainsi devenue un continuel défi. Les environnements hostiles caractérisés par des conditions extrêmes de hautes températures affectent le bon fonctionnement des systèmes. Pour les composants de stockage de données, la température est considérée comme une menace pour la fiabilité. Le développement de techniques de suivi et de contrôle devient ainsi essentiel afin de garantir la fiabilité des mémoires volatiles et non volatiles. Dans le cadre de ma thèse, je me suis intéressée à deux types de mémoires : les mémoires NAND Flash et les mémoires SRAM. Pour contrôler les effets de la température sur les mémoires Flash, une solution basée sur l'utilisation d'un timer a été proposée afin de réduire la fréquence de rafraîchissement tout en continuant à garantir l'intégrité de l'information stockée. De plus, une méthode statistique et une approximation calculatoire basées sur des opérations de vérification périodique ont été proposées afin d'améliorer le taux d'erreurs (RBER) tolérables dans des SSDs de types Entreprise à base de mémoires Flash. Enfin, pour les mémoires SRAM, l'effet de la température sur la vulnérabilité par rapport aux événements singuliers (SEU) a été étudiée. Une étude comparative sur l'apparition des SEU a été menée avec différentes températures pour des cellules standards 6T-SRAM et des cellules de stockage durcies (DICE).

Mots clés: Fiabilité, Contrôle, Température, Mémoires Flash, Rétention, RBER, Rafraîchissement, Mémoires SRAM, DICE, Événements singuliers

Monitoring of the effects of temperature on CMOS memories

Abstract: With the constant increase of microelectronic systems complexity and the continual scaling of transistors, reliability remains one of the main challenges. Harsh environments, with extreme conditions of high temperature and thermal cycling, alter the proper functioning of systems. For data storage devices, high temperature is considered as a main reliability threat. Therefore, it becomes essential to develop monitoring techniques to guarantee the reliability of volatile and non-volatile memories over an entire range of operating temperature. In the frame of this thesis, I focus my studies on two types of memories: NAND Flash memories and SRAMs. To monitor the effects of temperature in NAND Flash memories, a timer-based solution is proposed in order to reduce the refresh frequency and continue to guarantee the integrity of data. In addition, statistical and computational approximation techniques based on periodic check operations are proposed in order to improve the tolerated Raw Bit Error Rate (RBER) in enterprise-class Flash based SSDs. Finally, for SRAM memories, the effect of temperature on Single Event Upset (SEU) sensitivity is studied. A comparative study on SEU occurrence under different temperatures is conducted for standard 6T-SRAM cells and hardened Dual Interlocked Storage Cells (DICE).

Keywords: Reliability, Monitoring, Temperature, Flash memories, Retention, RBER, Refresh, SRAM memories, DICE, SEU

Acknowledgments

This thesis work has been conducted in the Laboratory of Reliability and Sensors Integration (LFIC), part of the CEA Laboratory of Systems and Technologies Integration (CEA-LIST).

Firstly, I would like to thank Eduardo Augusto BEZERRA and Alberto BOSIO who made me the honor to be the reviewers of my PhD thesis. I would also like to thank Marco OTTAVI, Frédéric WROBEL, Bruno ROUZEYRE and Hassen AZIZA who accepted to be part of my jury.

I would like to express my gratitude to Thierry COLETTE and Jean-René LEQUEPEYS, heads to the Department of Architecture, Design and Embedded Software (DACLE) for offering me the opportunity to conduct my thesis work in good conditions in their unit.

My sincere thanks also go to Antoine DUPRET and Tanguy SASSOLAS, heads of the LFIC, who provided me the opportunity to join their team as a PhD student, and who gave me access to the laboratory and research facilities.

I would like to express my sincere gratitude to my thesis director Luigi DILILLO for his crucial support, his insightful knowledge and trust and without whom I would not have achieved this work. I would also like to thank him for his warm welcome each time he received me in the Laboratory of Informatics, Robotics and Microelectronics of Montpellier (LIRMM).

Besides my thesis director, I would like to thank my advisor Valentin GHERMAN for his great help, his availability and his guidance. I would also like to thank him for all the rich scientific discussions we had which made me question my work and become a better researcher.

My sincere thanks goes to my mentor Jean-Marc Armani who provided me with valuable advice, crucial support and trust during the last two years. His guidance and great kindness helped me to overcome the difficulties during this research period and in the writing of this thesis.

I would also like to thank my fellow LFIC labmates for their support and help and for all the stimulating discussions we had. In particular, I am very grateful to Jaume, Esteban, Cyril, Mariem and Wafa for the valuable pieces of advice they gave me especially during this last year. I would also like to thank former labmates who left LFIC for other adventures and who were crucial to my work experience in the laboratory. I am grateful to Baptiste, Arnaud, Clément and Daniel for all the nice moments we shared and pleasant atmosphere they managed to bring to LFIC.

A special thanks goes to Emmanuel from LCE for his cheerfulness, his positive attitude and help. My sincere gratitude goes to Dzila my office labmate, but above all my friend and my confident, who always found the words to cheer me up, who was there for all my ups and downs and whose presence in the LFIC brought joy into my heart.

I mainly owe this achievement to my parents Leila and Jalel and to my sisters Asma and Eya who never doubted in my capacities or stopped encouraging me and who supported me physically and morally in all the crucial times of my life.

Finally, I would like to thank Thibaut for believing in me even when I did not, for his support, his great patience and joyful music. He knew how to cheer me up and I learned to rediscover life through his innocent gaze. Now the world is a better place.

Contents

List of Figures

List of Tables

Introduction

The continuous demand for electronic and microelectronic systems with higher performance and low power pushes the manufacturers towards technology limits and motivates the search for new physical and architectural solutions. In this context, reliability, which is defined as the probability that a given device or system works properly, is becoming an important matter [1].

Reliability requirements evolve due to the transistor shrinking and the development of emerging technologies based on new materials and technological processes. The interest in reliability is furthermore justified for applications where electronic components operate in harsh environments. These environments present extreme conditions of temperature and vibrations as is the case for automotive applications and oil & gas industries. Furthermore, in aerospace and military domains, electronic equipment may be exposed to ionizing radiation. These harsh conditions increase the probability of failures occurrence. For example, the exposure to high temperatures or to long and repetitive thermal cycles causes electric parameters degradation and physical properties of materials alteration, which results in components deterioration. The International Technology Roadmap for Semiconductors (ITRS), a report published yearly by a group of leading chip manufacturing organizations, evolves to match the needs of the semiconductor industry. In the 2013 edition of ITRS, reliability is referred to as a near-term and long-term challenge for CMOS logic devices, Dynamic Random Access Memories (DRAM) and non-volatile RAM [17].

The automotive industry is the one of the most concerned by exposure of electronic systems to high temperatures. Automotive standards fix the range of ambient operating temperatures from $-40\,^{\circ}\mathrm{C}$ to $125\,^{\circ}\mathrm{C}$. The upper limit may be easily exceeded, depending on the considered location, as is the case for electronic components mounted under the hood [18]. Oil & gas industries are also concerned by high temperature constraints. In fact, drilling operations are usually done at temperatures ranging from $150\,^{\circ}\mathrm{C}$ to $175\,^{\circ}\mathrm{C}$. Moreover, the decrease of natural reserves along with the technology advances pushed the drilling to greater depths with, as a direct consequence, well temperatures that exceed $200\,^{\circ}\mathrm{C}$ [18]. For aerospace applications, electronic control systems are no longer centralized but more distributed and are, in some cases, in contact with engines. This reduces the interconnection complexity but results in a working temperature going from $-55\,^{\circ}\mathrm{C}$ to $200\,^{\circ}\mathrm{C}$ [18].

These industries must insure the proper functioning of their electronic components despite the constraint of high temperature. Reliable data acquisition and storage constitute an important basis for components accurate operations. Volatile or non-volatile data storage mediums should be able to store data correctly through the entire range of operating temperature. For example, the *First High Temperature Electronics Products Survey* of Sandia National Laboratories indicated the industry players interest in data acquisition and storage, particularly in non-volatile memories as Flash memories [19]. Currently, the automotive industry shows a great need for embedded non-volatile memories with high reliability for an extended operating temperature range [20]. As for volatile memories, Static Random Access memories (SRAM) are widely used in integrated circuits

due to their speed and compatibility with standard logic processes. In addition, on-chip SRAM are one of the most important components for high performance system-on-chip (SoC) designed to operate at high temperatures [21][22].

Given all these elements, the target of this thesis is the study of the temperature impact on two types of memories: NAND Flash memories and SRAM. The effects of temperature on reliability are considered with an emphasis on retention for NAND Flash. Statistical and computational solutions aiming to improve the error rate in these memories have also been proposed. In SRAM, the effects of temperature on reliability are considered with an emphasis on resilience to Single Event Upsets (SEU).

This thesis is organized as follows. In the first chapter, an overview of reliability fundamentals and metrics is introduced. It is followed by details on the temperature effects on semiconductors with an emphasis on the electric parameters that impact reliability. Then, a general introduction on memories is given. For NAND Flash memories, the structure and operation are explained. For SRAM, the same study is done and is followed by an overview of radiation effects on integrated circuits.

In the second chapter, state-of-the-art techniques for reliability improvement in NAND Flash and SRAM are exposed. For NAND Flash, data retention and cycling endurance metrics are introduced and followed by a list of monitoring techniques based on the use of error correction codes, read operation optimization and data refresh. For SRAM, radiation effects are explained with emphasis on soft errors. State-of-the-art techniques for radiation effects resilience are finally given.

In the third chapter, the temperature effects on the chosen non-volatile memories, i.e. the NAND Flash memories, are addressed. Here, a timer-based solution is proposed to monitor the effect of temperature on the data retention capability. The timer purpose and working principle are first explained and compared to already existing solutions. The structure of the timer is then detailed. Simulation results related to the improvement of refresh frequency are also given. A final section explains the possible use of the timer with timestamps for warning triggering.

In the fourth chapter, the study of retention in NAND Flash memories is pushed further. Here, a statistical method and a computational approximation are proposed in order to improve the tolerated error rate in NAND Flash-based Solid State Drives (SSD). The two methods are explained with details on the used algorithms. The obtained results related to the error rate and average storage time improvements are also reported. Read and write times reduction resulting from the computational approximation method are detailed in the last section of this chapter.

In the fifth chapter, the temperature effects on the chosen volatile memories, i.e. SRAM, are addressed. The combined effects of thermal variations and radiation on SRAM are studied. A comparative study on the occurrence of single event upsets is made for two types of SRAM cells: the classic 6-transistor (6T) cell and the hardened Dual Interlocked Storage Cell (DICE). Here, the current pulses resulting from an ionizing particle strike on the sensitive nodes of the two cells are modeled. Simulation results on the evolution of the critical charge that provokes a bit flip are reported. Final remarks and conclusions are given at the end of this thesis.

Chapter 1

Overview of the target memories

In this chapter, we start by the fundamentals of device and systems reliability. A general study of the temperature effects on the electric parameters of Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFET) is then given. A short overview of the existing memory types is also presented. In particular, we focus on the two memory types studied in this thesis: NAND Flash and SRAM. Considering the NAND Flash memory, we describe its structure and detail its operation. A comparison between Single Level Cells (SLC) and Multi Level Cells (MLC) is also given. For SRAM, the conventional 6 transistor (6T) structure and its operation are explained. In addition, details on the cell stability are introduced with emphasis on noise margin. Finally, the causes of integrated circuits sensitivity to radiations are listed.

1.1 Reliability fundamentals

Over the last decade, the complexity of digital systems has substantially increased due to very large scale integration (VLSI) on chips and technology shrinking. Reliability issues came along and became a concern for both designers and customers. The probability that a system or device performs its intended function without failure under specific environmental conditions and for a specified duration highly depends on the used design and the quality of the components. In this section, we give an overview of device and systems reliability metrics.

1.1.1 Reliability and failure rate

In the sequel, reliability metrics are defined for a population of N identical components or devices which are considered under the same stress conditions during a given period of time t. Stress conditions are caused by environmental parameters such as temperature, humidity, shock or mechanical stress. Consider that $F(t)$ is the number of components that fail to function correctly after the stress process. $S(t)$ is the number of components that resist to stress conditions and continue to function correctly after t. Reliability $R(t)$ is defined as the probability of survival of components to stress conditions and is expressed as follows [1]:

$$R(t) = \frac{S(t)}{N} \tag{1.1}$$

The probability of components failure, also called unreliability is expressed as follows [1]:

$$UR(t) = \frac{F(t)}{N} \tag{1.2}$$

The failure rate $FR(t)$ is by definition the number of components failing per unit of time over the number of surviving components after a period of time t [1]:

$$FR(t) = \frac{1}{S(t)} \frac{dF(t)}{dt} \tag{1.3}$$

The *bathtub curve* as shown in Fig. 1.1 gives the evolution of the failure rate with time. This curve has been proven adequate to describe the reliability of electronic components and devices. It displays three types of variation zones:

- *Infant mortality period* which is a starting zone of high failure rate. Design issues or lack of quality control result in defective components when first put in operation. These defects can be minimized by *burn-in* tests where devices are exposed to temperature stress conditions close to the intended operating conditions. In case of devices survival after this test, they are released for actual use and this drastically decreases the *infant mortality probability.*
- *Useful life period* which is characterized by a constant failure rate. In this zone, failures appear randomly in time as they are generally not related to initial manufacturing or design issues. The failures in this period are the most critical as they occur in the useful life period of the equipment.
- *Wear-out period* where failure rate increases as a result of components aging and deterioration caused by repetitive use.

In case the *useful life period* is considered where the failure rate is constant and equal to λ, the following equation can be deduced using (1.3) and the fact that $R(t) + UR(t) = 1$ [1]:

$$\lambda \int_0^t \mathrm{d}x = -\int_1^{R(t)} \frac{\mathrm{d}R}{R} \tag{1.4}$$

and considering the limits of the integration, we get [1]:

$$R(t) = e^{-\lambda t} \tag{1.5}$$

1.1.2 Reliability and Mean-Time-To-Failure

Mean-Time-To-Failure (MTTF) is the average period of time during which a system runs correctly before the occurrence of the first failure. In practical use, it is more adequate to use MTTF than $R(t)$ that gives different values for different operating times. MTTF is given by the following equation [1]:

$$MTTF = \int_0^\infty t \left(-\frac{dR(t)}{dt} \right) \mathrm{d}t \tag{1.6}$$

For a constant failure rate (4.1), MTTF can be written as follows [1]:

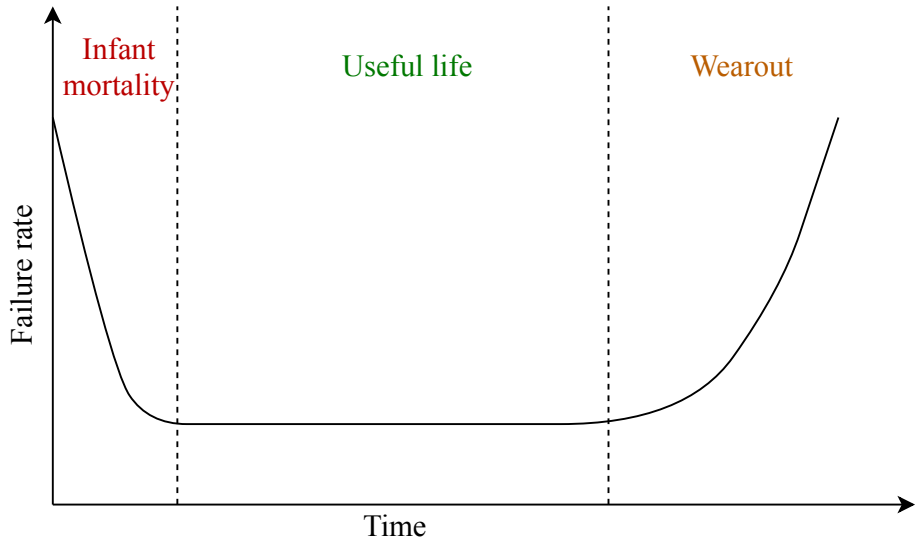

Figure 1.1: Failure rate variation with time [1]

$$MTTF = \frac{1}{\lambda} \tag{1.7}$$

Therefore, the relation between reliability and MTTF can be expressed as follows [1]:

$$R(t) = e^{-\frac{t}{MTTF}} \tag{1.8}$$

1.2 Temperature effects on semiconductors

Temperature variations at device and circuit levels affect the speed and power in semiconductors. In MOSFETs, temperature has a strong impact on electric parameters that are related to reliability. In this section, an overview of these effects is given and followed by details on electromigration in semiconductor-based devices.

1.2.1 MOSFET threshold voltage

The threshold voltage in a MOSFET is described as follows [23]:

$$V_t = V_{FB} + 2\phi_F + \gamma\sqrt{2\phi_F} \tag{1.9}$$

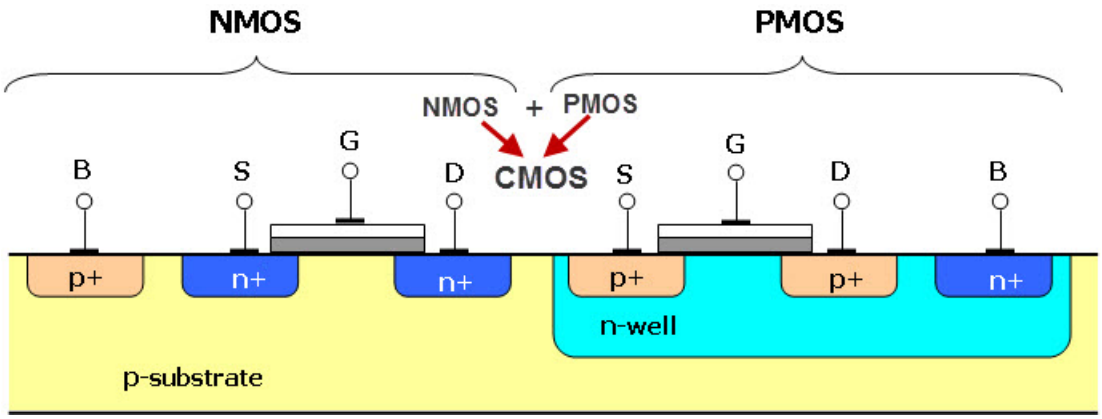

Figure 1.2: NMOS and PMOS structures [2]

where V_{FB} is the flat band voltage, $\phi_F = \phi_T \ln(\frac{N_A}{n_i})$ is the Fermi energy with the thermal voltage $\phi_T = \frac{kT}{q}$, N_A the substrate doping concentration, n_i the intrinsic carrier concentration of Si and γ is a body effect parameter. The flat band voltage is defined by $V_{FB} = \phi_{gs} - \frac{Q_{ss}}{C_{ox}}$ where $\phi_{gs} = \phi_T \ln(\frac{N_A N_G}{{n_i}^2})$ is the gate-substrate contact potential, N_G the gate doping concentration, Q_{ss} the surface charge density and C_{ox} the oxide capacitance.

In the threshold voltage expression, the parameters depending on temperature are ϕ_{gs} and ϕ_F and this dependence may be written as follows [24]:

$$\frac{\partial V_t}{\partial T} = \frac{\partial \phi_{gs}}{\partial T} + 2\frac{\partial \phi_F}{\partial T} + \frac{\gamma}{\sqrt{2\phi_F}}\frac{\partial \phi_F}{\partial T} \tag{1.10}$$

Commonly, a MOSFET is modeled to have a threshold voltage decreasing with temperature increase. This can result in reliability problems in NAND Flash memories if V_t is significantly decreased by temperature.

1.2.2 MOSFET drain current

The behavior of drain current (I_d) in a MOSFET varies with temperature. At a constant drain to source voltage (V_{ds}) and depending on the applied gate voltage (V_{gs}), a MOSFET is either in ON or OFF state. When V_{gs} is above the threshold voltage V_t and the drain is positively biased, a n-channel MOSFET is in ON state and a current I_{on} flows between the drain and source. This current is approximated by [25]:

$$I_{on} = \mu C_{ox} \frac{W_g}{L_g} (V_{gs} - V_t)^2 \tag{1.11}$$

where W_g and L_g are respectively transistor gate width and length, C_{ox} is the oxide capacitance and μ the electron mobility in the channel. For a fixed value of V_{gs} above V_t, I_{on} decreases with temperature due to the decreasing mobility and the slight decrease of V_t [25][26]. By decreasing V_{gs} below V_t, the current flow in the channel is normally turned off but parasitic electrons create a *subthreshold leakage current* I_{sub} highly dependent on temperature.

1.2.3 MOSFET Leakage current

Subthreshold leakage current I_{sub}, between the source and drain of a MOSFET in the subthreshold region, has an exponential dependence on temperature. The subthreshold leakage current is represented by the Shockley diode model [24][27]:

$$I_{sub} = I_0 \left(e^{\frac{V_{ds}}{\phi_T}} - 1 \right) \tag{1.12}$$

where I_0 is the reverse saturation current.

The exponential temperature dependence of I_{sub} comes from the expression of the thermal voltage $\phi_T = \frac{kT}{q}$. Also, it should be noted that the subthreshold leakage current is predominant over other forms of leakage current in a MOSFET. This exponential temperature dependence of I_{sub} threatens the reliability of NAND Flash memories and SRAM operating at high temperatures.

1.2.4 Electromigration

High currents flowing through wires can eventually damage them, and electromigration is the main failure mode. It is caused by the shift of atoms in material under the effect of high-energy electrons. This failure mechanism has an important impact in the case of high current densities where the atoms moving down a wire narrow its width and further increase current density. The narrowing of the wire increases the wire resistance and may result in a delayed signal propagation and eventually a functional failure.

As electromigration can result in system failures, its impact on reliability is measured via the MTTF defined by Black's equation [2][28]:

$$MTTF = A_j J^{-n_j} e^{\frac{E_a}{kT}} \tag{1.13}$$

where A_j is a constant related to the wire cross section area, J is the current density, n_j a constant scaling factor and E_a the activation energy corresponding to the failure mechanism.

1.3 General introduction to memories

Memories are key components for computer and electronic systems since they are used for both data and program instructions storage. The need for different storage capacities depends on the application field. For example, data centers are in the order of Petabytes of storage capacity, personal computers are in the order of hundreds of Megabytes to Gigabytes of memory and embedded systems tend to have smaller amounts of memory, from hundreds of Kilobytes to Megabytes. Different types of memories are used in electronic systems. Each type corresponds to a different technology and has a different storage capacity, power requirements, latency and price. The volatility and the data access mode can also vary from one type of memory to another. For these reasons, each electronic system relies on a mix of the different types of memories.

The pyramidal structure of the computer memory hierarchy is illustrated in Fig. 1.3. On top of the pyramid are central processing unit (CPU) registers that hold words retrieved from CPU caches. Both registers and caches are fast, with smaller storage capacity and higher cost per byte stored. For computer systems, memory is divided in two main groups: the main memory and the mass storage memory. The main memory communicates with the CPU and is usually made out of Random Access Memories (RAM). RAMs have a small access time, an average storage capacity and they are priced reasonably. They are read/write memories accessed for both data storage and retrieval, in contrast to read-only memories (ROMs) only accessed for reading data. RAMs also come in two varieties: Static RAM (SRAM) and Dynamic RAM (DRAM). Mass storage requires relatively cheap data storage devices with large capacities and lower speed. These requirements are met by hard drives using magnetic storage of up to Terabytes of data, and more recently by NAND Flash-based Solid State Drives (SSD). Flash memories are non-volatile, i.e. able to retain data when the power is off, but SRAM and DRAM are both volatile. This gives another classification of memories based on their volatility. The type of data stored in SRAM, DRAM or Flash memories depends on their features and requires some compromises. SRAM are faster than DRAM but cost more and come in less density given a fixed area. Flash memories are non-volatile and less expensive than volatile RAM but they have higher access times [29].

Scaling affects these different types of memories. As transistors shrink, switching speed increases and manufacturing cost decreases. However, scaling introduces new problems related to complexity increase and exacerbation of reliability issues. Moreover, memory designers need to ensure the possible adaptation of existing processes to size reduction. In this context, the Semiconductor Industry Association (SIA) develops the International Technology Roadmap for Semiconductors (ITRS). This roadmap combines the efforts of different companies and researchers to develop compatible process steps and anticipate and challenges related to scaling [30].

The integration limit of Flash memories approaches and new types of emerging memories are starting to replace them [31]. The non-volatility of Flash, the speed of SRAM and the density of DRAM features can be balanced in emerging non-volatile memories such as Spin-Transfer Torque Random Access Memory (STT-RAM), Ferroelectric Random Access memory (FeRAM), Phase-Change Memory (PCM), and Resistive Random Access Memory (RRAM).

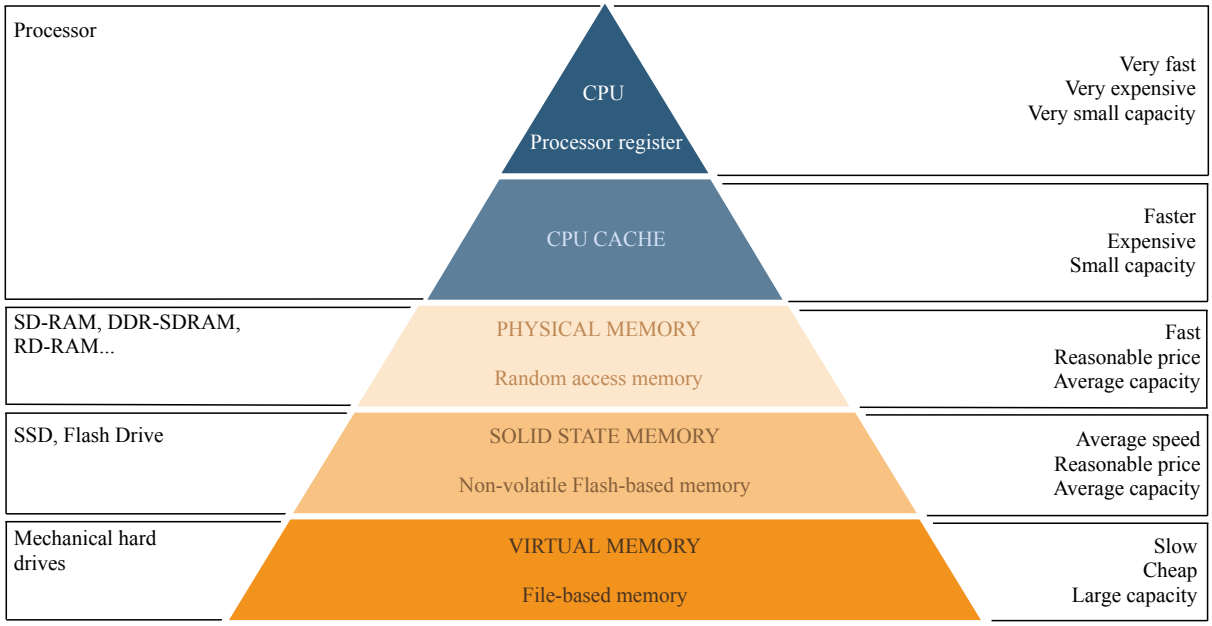

Figure 1.3: Simplified computer memory hierarchy

In this report, we focus on NAND Flash memories and SRAM.

1.4 NAND Flash memories

Among the different types of memories, NAND Flash technology offers a cost effective solution for applications requiring high density and solid state storage. NAND Flash, together with NOR Flash memories, dominate the market of mass storage non-volatile memories [32].

The first NOR Flash memories were commercialized by Intel in 1988 as a non-volatile storage medium for program codes based on the invention of Flash memory by Dr. Fujio Masuoka from Toshiba. They were then used in the first Flash cards and non-volatile solid state drives. Toshiba then introduced the NAND Flash memories in 1989 characterized by their lower cost per bit and their faster access time [3].

NOR Flash memories have a parallel array architecture that provides a direct access to cells and guarantees a better random performance compared to NAND Flash which have a serial array architecture. The internal structure of NAND Flash, explained in the following, provides a smaller area and a faster write ability compared to NOR Flash, but also results in a degraded access randomness. The lower cost per bit of NAND Flash makes it suitable for low cost mass storage.

In this section, we first introduce the structure of a NAND Flash memory. Then, we detail the read, write and erase operations. A comparison between single level cells and multilevel cells is

presented at the end of this section.

1.4.1 NAND Flash structure

A NAND Flash memory cell is made of a floating gate transistor. This is a particular type of MOS transistors characterized by the presence of two overlapping gates instead of one (Fig. 1.4) [3][31][33][34]:

- *A control gate* (CG) with a contact providing a gate terminal as in a standard MOS transistor.
- *A floating gate* (FG) where electrons are stored when the corresponding cell is programmed. They generate a counter electric field partially screening the one produced by the CG. The FG is surrounded by two layers of oxide:
  - *A tunnel oxide* underneath the FG, which is a weak insulator through which electrons tunnel between the FG and the substrate in case of a strong enough voltage differential.
  - *An interpoly* or *blocking oxide* above the FG, which is a strong insulator that separates the two gates and prevents electrons from passing through the FG to the CG.

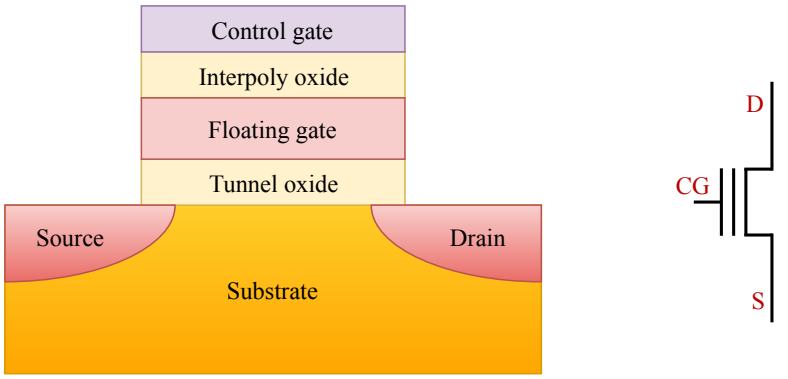

Figure 1.4: Schematic of a floating gate transistor and its electric symbol [3]

By applying appropriate voltages on the transistor inputs, electrons can be injected to or extracted from the FG which results in a program or erase operation of the FG transistor. When injected in the FG, electrons remain trapped. Here, a FG transistor is considered in the case of a Single Level Cell (SLC) which is able to store a single bit. The program and erase operations change the transistor threshold voltage and this results in two logic states of the cell: 0 and 1. A programmed cell with a higher threshold voltage corresponds to a logic 0 and an erased cell with a lower threshold voltage corresponds to a logic 1 [3].

It needs to be noted that environmental and manufacturing sources of variations have an influence on the characteristics of NAND Flash memories as it is the case of any other microelectronic

component. The variation sources are: process variation, supply voltage and temperature also known as PVT. These variations are modeled by uniform or normal statistical distributions. Process variations are usually modeled with a Gaussian distribution [2]. In the case of NAND Flash, process variations have an impact on the threshold voltage of FG transistors and the Gaussian distribution is used as an accurate model for Flash cells threshold voltage distributions [10][35]. Fig. 1.5 shows the threshold voltage (V_t) distribution in the case of an SLC.

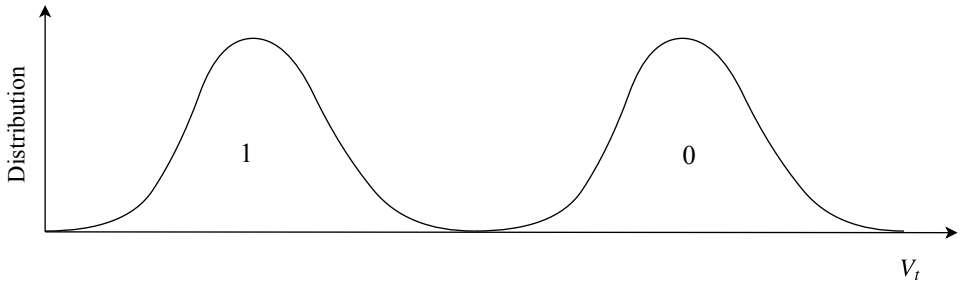

Figure 1.5: Threshold voltage distribution of an SLC [3][4]

Memory cells are organized in a matrix form in order to minimize the total area. Depending on how FG transistors are connected in the Flash memory array, NOR and NAND Flash memories can be distinguished. Fig. 1.6 shows a comparison of NAND and NOR Flash memory cells arrangement.

NAND strings are composed of FG transistors connected in series and placed between two selection transistors: a Source Selection Transistor (SST) that connects to the Source Line (SL) and a Drain Selection Transistor (DST) that connects to the bitline (BL) (Fig. 1.6 and Fig. 1.7).

Fig. 1.7 shows a NAND array composed of a series of blocks, a block being the smallest erasable entity. Each block is composed of a group of NAND strings that share the same wordlines (WL) which justifies the simultaneous erase of an entire block.

In a NAND Array, there may be two types of bitlines: even BL and odd BL as shown in Fig. 1.7. Memory cells controlled by the same WL form a logical page and each block contains multiple pages. A page is the smallest addressable unit in read and write operations. Even and odd cells controlled respectively by even and odd BLs, form two different logical pages. Each page is composed of main and spare cells. Main cells are for data storage and spare cells are usually used to store redundant information necessary for the implementation of error correction codes.

Apart from the memory array, a NAND Flash memory needs additional circuitry for read, write and erase operations. A row decoder commands the WLs belonging to a selected NAND string for a read, write or erase operation. BLs are connected to sense amplifiers that convert the current that goes through a memory cell during a read operation into a digital value. Charge pumps, voltage regulators, logic circuits and redundancy structures are also needed for the proper functioning of a NAND Flash memory.

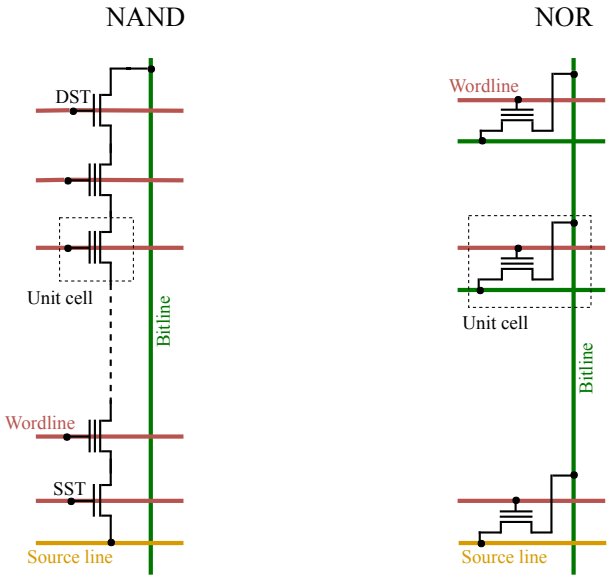

Figure 1.6: Comparison of NAND and NOR Flash cells arrangement [5]

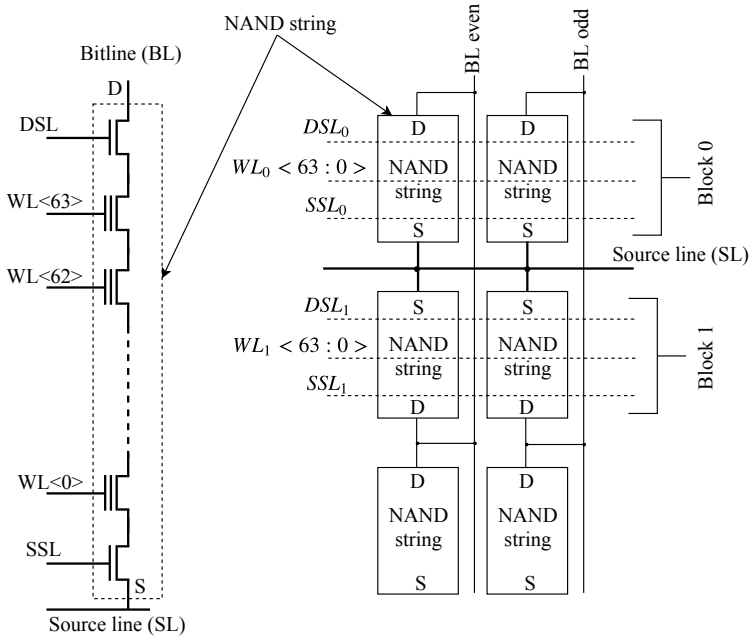

Figure 1.7: Schematics of NAND string (left) and NAND array (right) [3]

1.4.2 NAND Flash operations

1.4.2.1 Read operation

A read operation of a NAND Flash memory is a conductivity test operated on the string corresponding to the cell that needs to be read. Charge presence or absence in the FG respectively inhibit or allow the string conductivity. The string current response is used to distinguish a programmed cell from an erased one.

The V_t distributions of Fig. 1.8 is considered in the case of single level cells [3]. The challenge with the read operation of a cell is that it requires the rest of the corresponding string to be conducting. This needs to be done independently from the state of the cells composing the rest of the string (programmed or erased). For the read operation, these cells need to be driven by a high voltage $V_{pass,R}$ in order to force their conductivity regardless of their states. This voltage is typically equal to 4–5V. For the read cell, the gate voltage V_{read} needs to be equal to 0V. Besides, a voltage differential is present between the source and drain of the read cell by the application of adequate voltages on the bitline and on the gates of DST and SST (Drain Selection Line (DSL) and Source Selection Line (SSL), respectively Fig. 1.7) belonging to the string of the read cell [3][8].

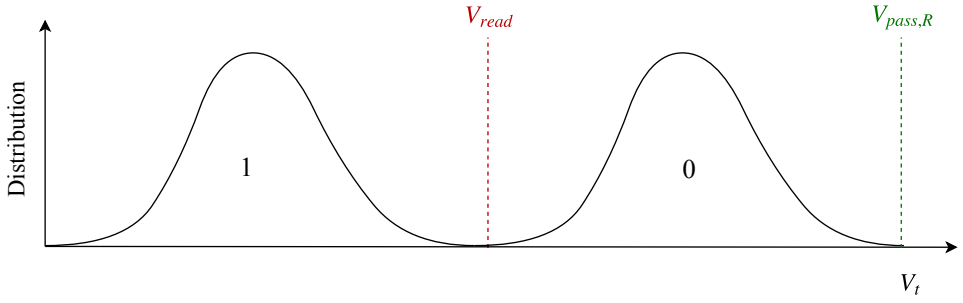

Figure 1.8: Threshold voltage distribution of an SLC with reference voltages [3]

If the read cell is erased, electrons are not present in the FG and this provides no interference between the CG and the substrate. The read cell conducts resulting in a current flowing through the whole string which is interpreted as a logic 1 by a sense amplifier. If the read cell is programmed, the charge on the floating gate will inhibit the low voltage applied on the control gate by the generation of a counter electric field that screens the electric field from the CG. The read cell will not conduct and the absence of current on the string will be interpreted as a logic 0 (Fig. 1.9).

Due to the organization of the NAND Flash array, a cell read operation is achieved by the application of different voltages on the BLs and WLs of the addressed block as it can be seen in Fig. 1.10.

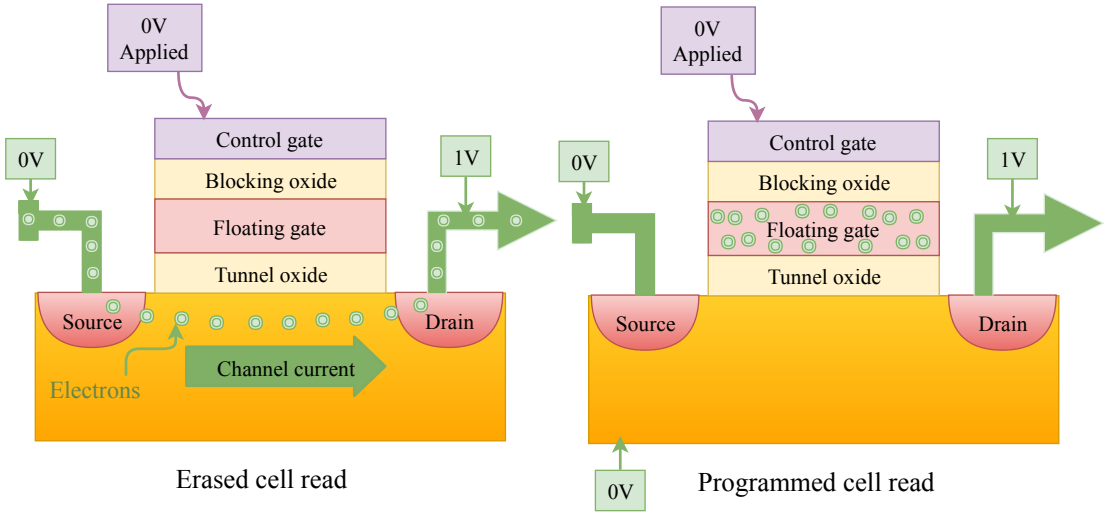

Figure 1.9: Read operation of an erased NAND cell (left) and a programmed NAND cell (right) [3]

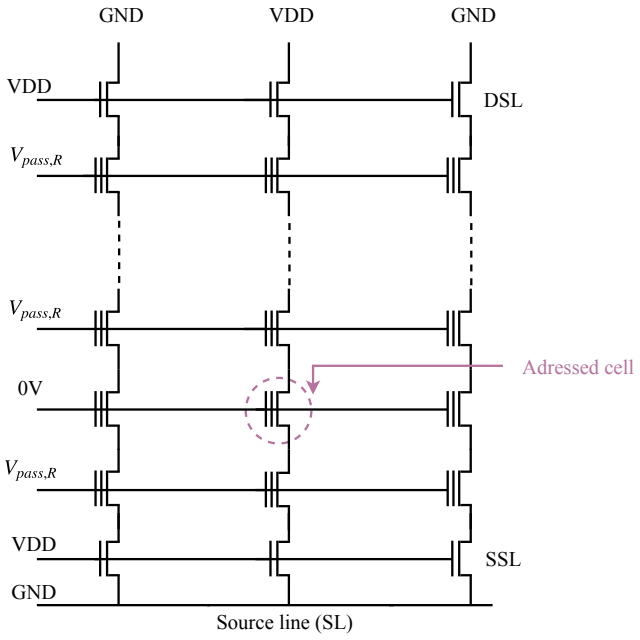

Figure 1.10: NAND block read [3]

1.4.2.2 Program operation

When programming a NAND cell, a high voltage is applied to the control gate. This results in a high voltage differential and a strong electric field across the gate oxide. With *Fowler-Nordheim* (FN) tunneling that exploits the quantum-effect of electron tunneling, electrons present in the semiconductor substrate will migrate to the FG through the tunnel oxide. But, they are stopped by the blocking oxide from tunneling to the CG. The stronger the electric field, the higher the injection probability and the number of electrons crossing the tunnel oxide. When the program operation is complete, electrons remain trapped in the FG as they do not have enough energy to tunnel back to the transistor channel. When they are present in the FG, electrons generate their own electric field which interferes with the tests of conductivity during read operations and indicate whether the cell is programmed or not (Fig. 1.11).

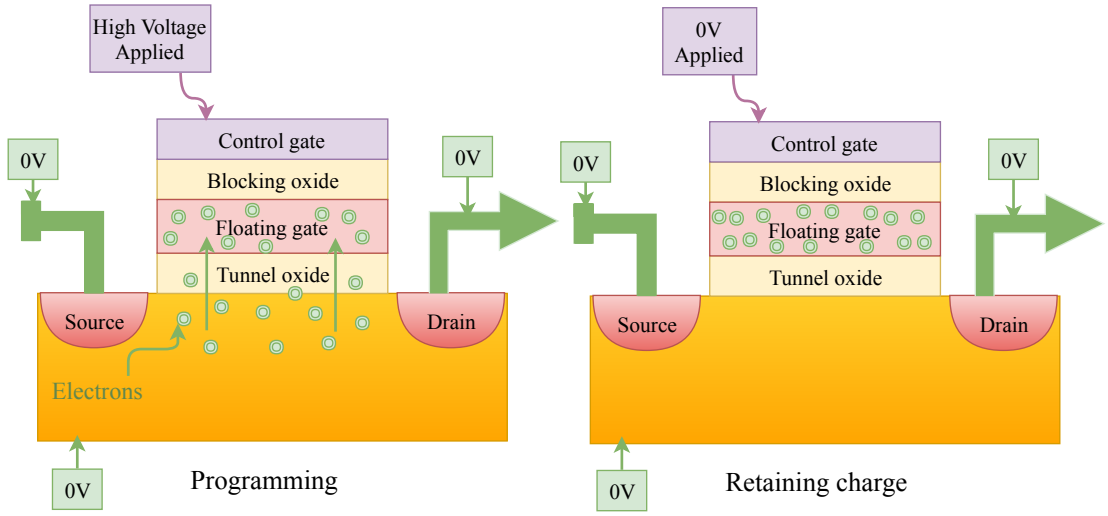

Figure 1.11: Programming (left) and retaining charge (right) in a NAND cell [3]

As in the read operation, different voltages need to be applied on the NAND array terminals to insure the programming of the target cell without modifying the neighboring ones. As shown in Fig. 1.12 and in order to trigger the injection of electrons into the FG, the following voltages need to be applied [3][8]:

- $V_{program}$ on the gate of the selected transistor to be programmed. This voltage need to be high enough (20–25V) to extract electrons from the substrate and tunnel them to the FG.
- Supply voltage (VDD) on DSL and ground (GND) on SSL.
- GND on the BL to be programmed.

- VDD on the other inhibited BLs and $V_{pass,P}$ on the unselected gates. $V_{pass,P}$ is a moderate voltage (8–10V) that, combined with the VDD applied to the inhibited BLs, results in a *self-boosting* mechanism preventing the cells sharing the same WL as the programmed one from going through an undesired program. This phenomenon is based on the creation of a capacitive coupling that boosts the potential of the unselected cells channel and thus results in a *tunneling inhibition*.

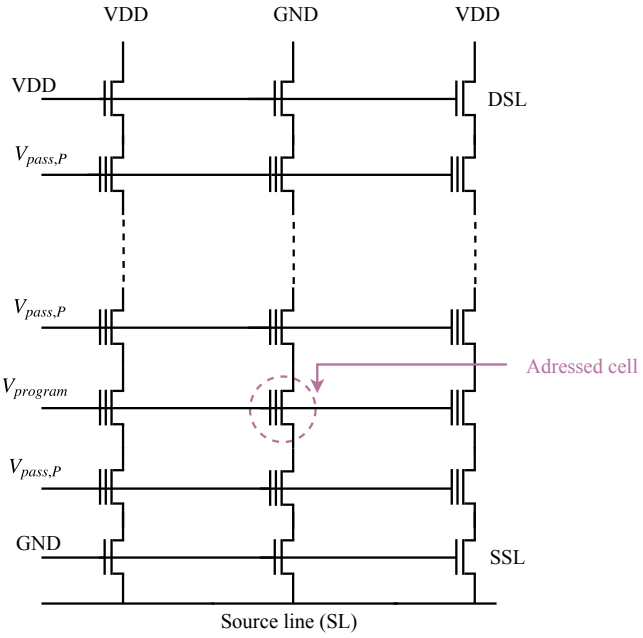

Figure 1.12: NAND block program [3]

1.4.2.3 Erase operation

Erase operation is based on reversing the *Fowler-Nordheim* mechanism used in program operations. To achieve the electrical erase, the wordlines of the block to be erased need to be biased at 0V and the substrate needs to be biased with a high voltage (18V). This creates a voltage differential that results in electrons tunneling back to the substrate through tunnel oxide and emptying the FG. To prevent the undesired erase of the unselected blocks that share the same p-well, their corresponding WLs are left floating. With the charged p-well, the floating WLs will see their potential rising due to capacitive coupling and this way, *Fowler-Nordheim* mechanism is inhibited in these blocks (Fig. 1.13) [3][8].

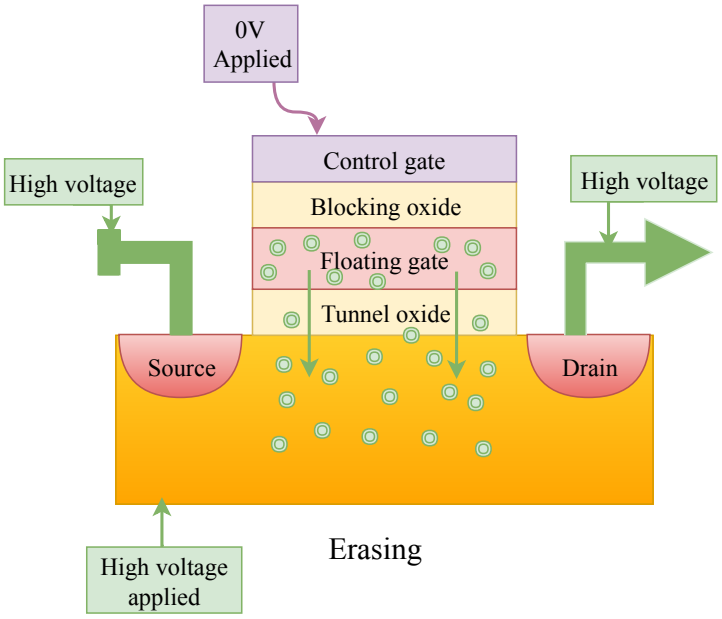

Figure 1.13: NAND cell erase [3]

1.4.3 Single vs. multiple level cells

Historically, NAND Flash memories were able to store a single bit per cell, thus the appellation Single Level Cell (SLC). Then, they evolved into Multi Level Cell (MLC) NAND Flash where two or more bits can be stored per cell. The storage of 2 bits per cell is achieved by adapting the amount of stored charge, so that four logic levels are available instead of two. Compared to the case of an SLC, the erased state (logic 11) and the fully programmed state (logic 00) are kept the same with a narrower V_t range (Fig. 1.14). The two additional programmed states (logic 10 and 01) correspond to a partially charged floating gate (Fig. 1.14). Three, four or more bits can be stored by cells by increasing the number of logic levels to 8, 16 or more. Fig. 1.14 also shows the case of a Triple Level Cell (TLC) able to store three bits per cell [3].

Multi level storage increases the storage capacity without increasing the process complexity. However, it relies on the ability to precisely control the amount of charge injected into the FG in order to set the threshold voltage distributions that correspond to the wanted number of logic levels. A cell able to store n bits should be operated with 2^n logic levels. A high program accuracy and more complex programming algorithms, such as Incremental Step Pulse Programming (ISPP) that programs floating gates iteratively and step by step by injecting small amounts of charge followed by a verify approach, are then necessary to have narrow V_t distributions and avoid the overlapping

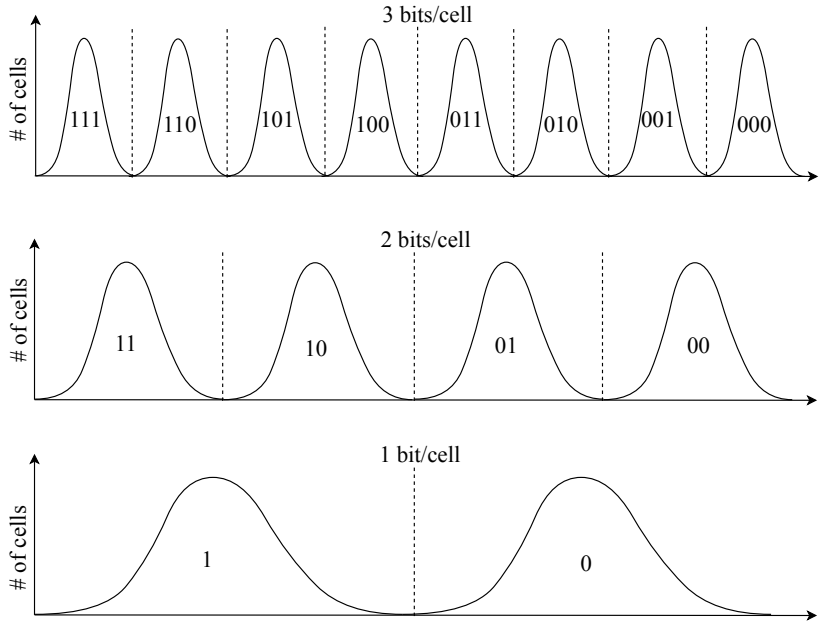

Figure 1.14: Multi level storage in NAND Flash memories [3]

resulting from the multiplication of logic levels. Reading operations are also more complex due to the number of read voltages increase and read margins reduction. Accurate and fast current sensing is then needed [34].

Improved reliability needs to be guaranteed as the higher capacity and lower cost per bit of multi level storage is compromised by degraded performance and endurance. In fact, multi level cells tend to wear out faster than SLC due to their higher sensitivity to physical changes in the tunnel oxide layer. Additional reliability problems are experienced due to the multiple variations of the CG voltage and the increased disturbance from neighboring cells [3].

Here, cells able to store only 2 bits are referred to as MLC. As it can be seen in Fig. 1.15, the comparison between SLC, MLC, 3-bit and 4-bit per cell memories show a reduction of endurance, retention, write and read performances and operating current with the increasing number of bits per cell. Erase speed is improved due to larger block sizes. Adequate memories are then chosen depending on the application type and the intended usage. For mass storage applications such as archiving, 4-bit per cell storage may be preferred. For read intensive consumer applications such as Flash memory cards, 3-bit per cell storage can be used. For applications with smaller write performance needs, MLC storage is sufficient. Besides, SLC storage is preferred for reliability demanding applications whether multi level storage is more used for mass storage consumer applications [3].

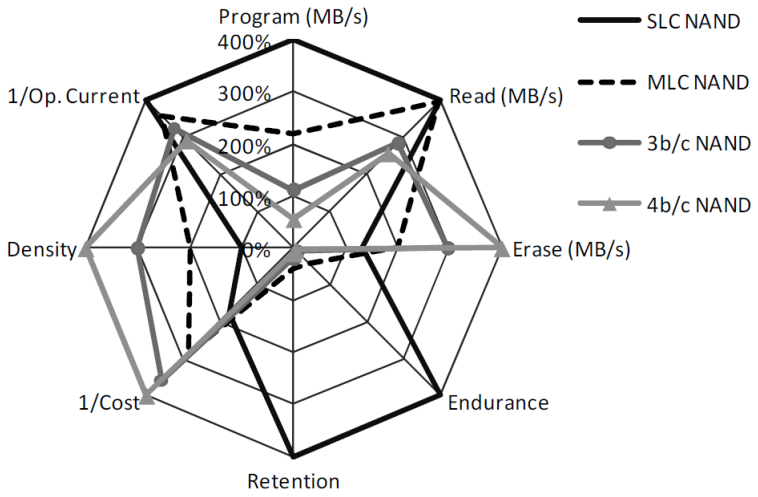

Figure 1.15: NAND storage cells characteristics [3]

1.5 SRAM memories

SRAM has long been a standard commodity-type product filling memory needs in different applications ranging from consumer electronics to high-end computers. Nowadays, it is mainly used in CPU register files, CPU caches and buffers. The first SRAM chip was released by Intel based on the invention of John Schmidt at Fairchild Semiconductor in 1969. The next year, the first DRAM memory chip was also released by Intel [36].

SRAM uses cells that maintain their state trough feedback, whereas DRAM cells store charge on a floating capacitor through an access transistor. Charges tend to leak in DRAM through the OFF-state access transistor. For this reason, DRAM cells must be refreshed by periodic read and rewrite operations, whereas SRAM cells retain their data as long as power is applied due to the internal feedback.

SRAM has different attractive properties: it is faster and easier to use than DRAM. It is also logically compatible with standard CMOS processes. In addition, technology scaling enabled SRAM size reduction and its dense integration on chips. On the other hand, this introduced problems related to the cost and leakage power consumption. Supply voltage control along with size reduction also made SRAM more vulnerable to soft errors and more precisely to Single Event Upsets (SEU).

In this section, we first introduce the structure of an SRAM with emphasis on the composition of a standard cell. Then, we detail the read and write operations. The fundamentals of cell stability and noise margin are also addressed. Generalities on radiation effects on integrated circuits are presented at the end of this chapter.

1.5.1 SRAM structure

An SRAM is mainly composed by storage cells able to hold data as long as power is applied, and a peripheral circuitry for writing and reading data stored into the cells. Many SRAM Cell designs exist and each involves a different number of transistors ranging from 4 to 12. Here, we consider a 6-transistor (6T) organization, which is the most common design for SRAM cells due to its low leakage and good compactness.

It should be noted that the operations provided by a 6T-SRAM cell can also be accomplished by a standard flip-flop, but a 6T-SRAM cell has the advantage of having an area smaller by an order of magnitude than a flip-flop [37]. This compactness feature is beneficial when memory cells dominate the area as in large RAM arrays. It also results in lower power dynamic consumption. However, compactness is achieved at the expense of a more complex peripheral circuitry [38].

A 6T-SRAM is composed of a pair of weak cross-connected inverters to hold the stored data and two NMOS access transistors enabling the read and write operations as it can be seen in Fig. 1.16. The pair of cross-coupled inverters enables the storage of a single bit, either a logic 1 or 0, set through the voltage difference forced on the bitlines couple, BIT and BITB. During the storage phase, small disturbances caused by noise or leakage can be corrected by the positive feedback of the pair of inverters. These disturbances may be increased by different operating and environmental conditions and eventually result in failures. The wordline signal, WORD in the 6T-SRAM circuit, drives the two NMOS access transistors (A1 and A2) enabling the charge transfer between the cell nodes Q and QB, respectively with BIT and BITB during write and read access cycles. The main challenge with an SRAM cell is to keep its area minimal while ensuring that the cell circuitry is sufficiently weak to be overpowered during a write access, but yet strong enough to keep the stored value correct and not erased during a read access.

For a 6T-SRAM array, cells are arranged in a matrix form with each cell being accessed for both read and write operations via BIT and BITB and through the access signal WORD. In the case of an SRAM array containing 2^n words of 2^m bits each (Fig. 1.17), bits are stored in 6T-cells via:

- The application of the adequate voltages on BIT and BITB depending on the value of the bit to be stored. Here, the column circuitry and column decoder are involved. It should be noted that column circuitry may contain amplifiers and buffers to sense data.
- The selection of the adequate row of cells to store data via the row decoder. The latter uses the received address to activate the adequate row by asserting the corresponding wordline through the signal WORD.

1.5.2 SRAM operations

1.5.2.1 Read operation

A bit stored in an SRAM cell is read by first precharging the two bitlines high and then leaving them both floating. The signal WORD is raised high and the two NMOS access transistors are

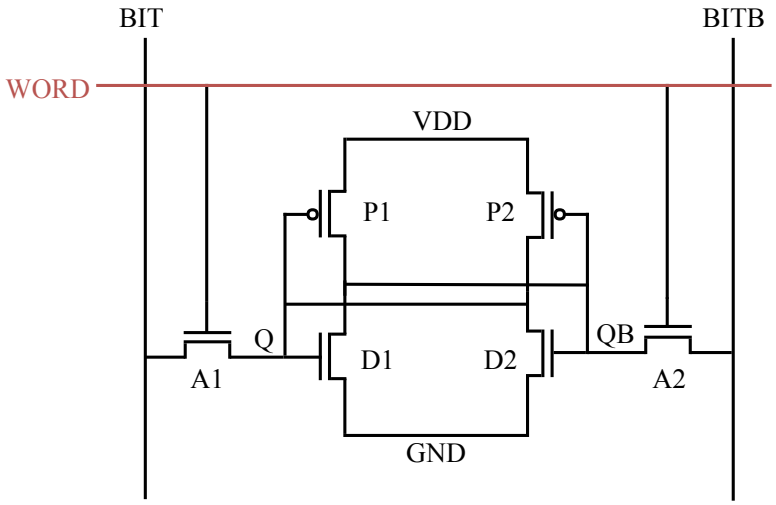

Figure 1.16: 6T-SRAM cell structure [2]

in ON state. This enables the cell nodes Q and QB to transfer the stored data to BIT and its complement on BITB.

Without loss of generality, Q is assumed to store a logic 0 (QB is at logic 1). When WORD is high, BIT is pulled down through A1 and D1 which are in ON state. While BIT is being pulled down, Q tends to slightly rise because of the current flowing through A1 but is ultimately held low by D1 (Fig.1.18). Here, a *read stability* constraint should be met by having D1 stronger than A1. Generally, this constraint is satisfied by fixing the W/L ratio of the transistors used in the pair of inverters such that the nodes Q and QB remain below the switching threshold of P2/D2 and P1/D1 inverters respectively [2][39].

1.5.2.2 Write operation

A write operation in a 6T-SRAM cell is done by driving the value to be written and its complement on BIT and BITB, respectively. The wordline WORD is then raised high which puts the two access transistors in ON state and enables the charge transfer between the cell nodes Q–QB, and BIT–BITB, respectively. The new data is then able to overpower the cross-connected inverters of the cell.

Without loss of generality, Q is assumed to be initially set to 0 and we intend to write a 1 in the cell. So, BIT needs to be raised high by a write driver. BITB is pulled down to ground and A1 and A2 are put in ON state through WORD rise. In fact, due to the *read stability* constraint, BIT is unable to raise Q high alone through A1 (A1 wheaker than D1). QB needs to be pulled low through A2. As P2 opposes this operation, P2 needs to be weaker than A2. This is known as *writability*

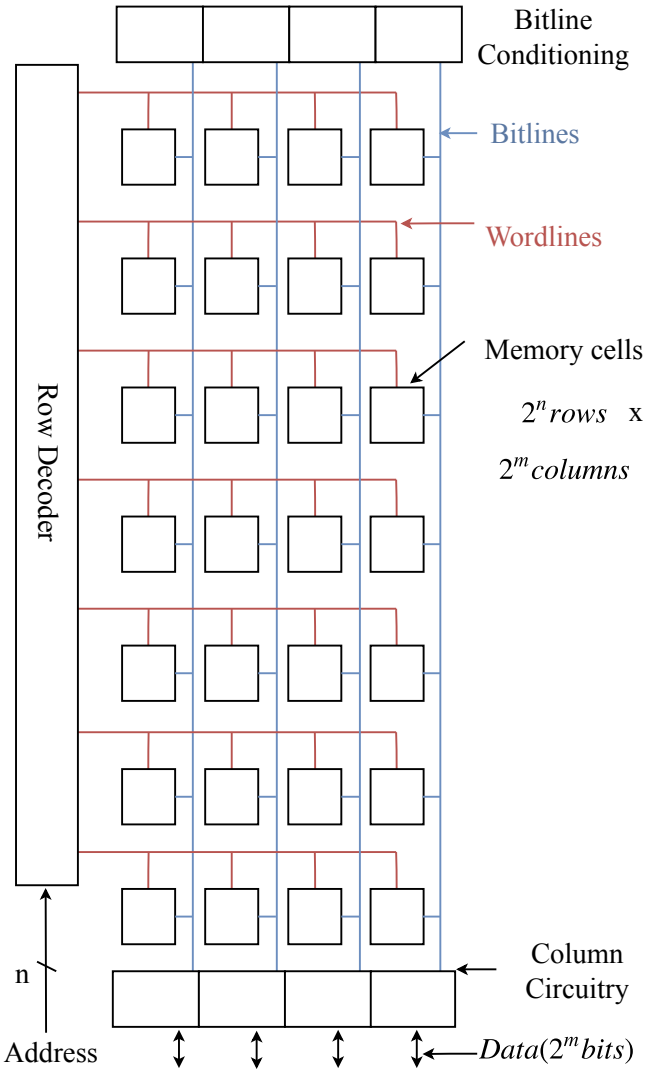

Figure 1.17: SRAM array architecture [2]

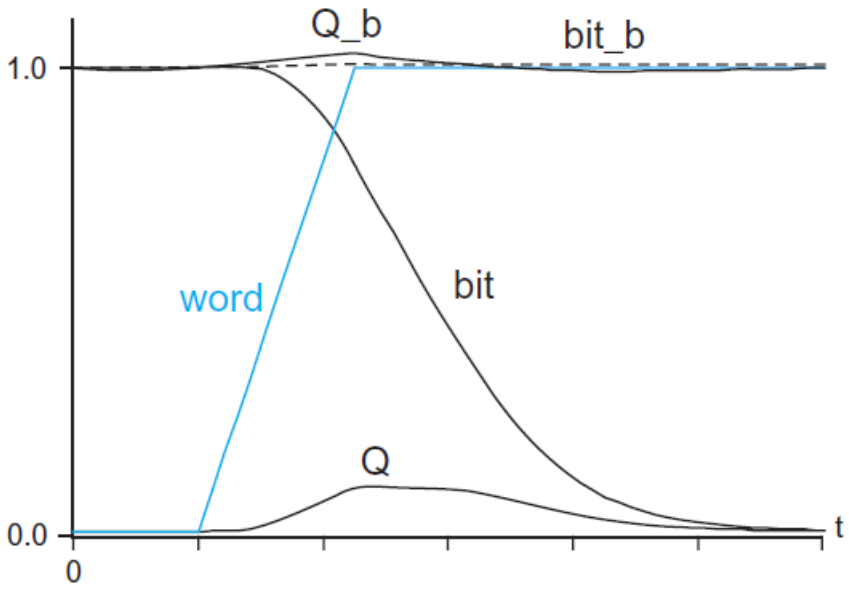

Figure 1.18: Read operation for 6T-SRAM cell [2]

constraint. QB being pulled low, this will put D1 in OFF state and P1 in ON state which will pull Q high as desired (Fig.1.19) [2][39].

1.5.2.3 Hold phase and cell stability

The 6T-SRAM cell is intended to hold the stored value in the pair of cross-connected inverters as long as the supply voltage is applied. During the hold phase, WORD is pulled low which puts the access transistors in OFF mode. This way, BIT and BITB are disconnected from the cell nodes Q and QB, respectively and the stored data remains unchanged in the cross-coupled inverters.

Due to *read stability* and *writability*, transistors must respect several ratio constraints. D1 and D2 must be the strongest among the 6 transistors of the SRAM cell, A1 and A2 must be of intermediate strength and P1 and P2 must be weak. Along with size and design constraints, SRAM cells must be able to function correctly despite voltage, temperature and process variations [40].

The determination of the cell noise margin in the different operating modes allows the quantification of hold, read and write margins that insure the stability and writability of the cell. In hold and read operations, two stable states are considered and the Static Noise Margin (SNM) qualifies the maximum amount of noise that can be withstood by the inputs of the pair of cross-connected inverters before a stable state is lost. During the write operation, only the written state matters and SNM measures the maximum noise that can be withstood before the second state is created.

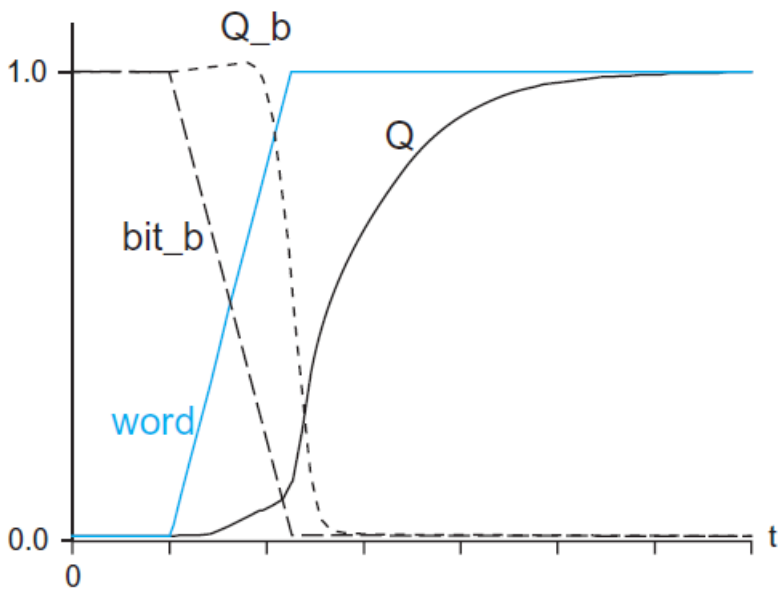

Figure 1.19: Write operation for 6T-SRAM cell [2]

Here, we consider the hold operation. To determine the hold margin, DC error sources (V_n) need to be applied to the internal data nodes of the SRAM cell as it can be seen in Fig. 1.20. Access transistors are not represented as they do not affect the circuit behavior. Hold SNM is deduced from a *butterfly curve* as shown in Fig. 1.21 [41].

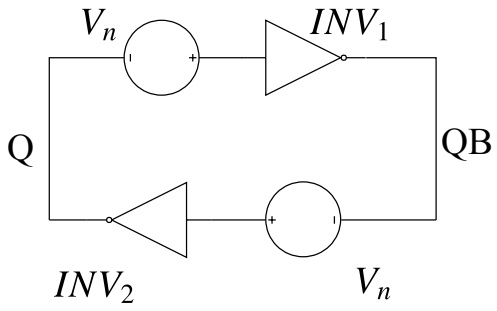

Figure 1.20: DC error sources in cross-coupled inverters for hold SNM calculation [2]

This curve is obtained by setting the initial value of the DC error source to 0V and plotting Q against QB and QB against Q when error source amplitude is varied. When the two inverters in the 6T-SRAM are identical, the butterfly curve exhibits a symmetry relative to the line Q=QB. When

error noise voltage increases, the stable states of Q and QB are eventually eliminated and the cell skips to the opposite state. The hold SNM is equal to the side length of the largest square that can be fitted in both loops of the butterfly curve. If the inverters are identical, high and low SNM are equal. Otherwise, the SNM is defined by the lowest of the two values. It should be noted that hold SNM increases with supply voltage (VDD) [42][43].

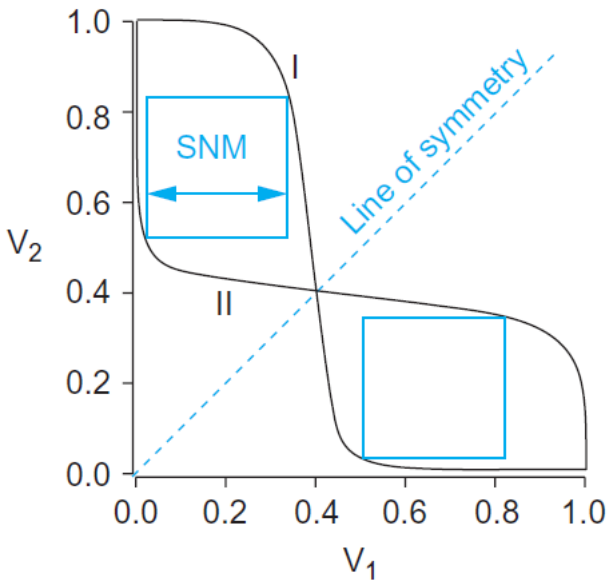

Figure 1.21: Butterfly curve for a 6T-SRAM cell [2]

1.5.3 Radiation effects on integrated circuits

Sensitive regions of microelectronic devices and circuits may be affected by natural radiation coming from space or in the terrestrial environment. With the constant transistor shrinking, the sensitivity to radiation effects is increasing and is becoming a major concern for reliability. Problems related to radiation are well known in the aerospace community. Interaction of particles like protons, heavy ions and neutrons with matter is considered as a main cause of Single Event Effects (SEE) in integrated circuits. Here, an overview of particles interaction with matter is given, followed by generalities on the effects of radiation on integrated circuits.

1.5.3.1 Particles interaction with matter

The main particles that interact with matter are: photons, neutrons and ionized particles as protons, alpha particles and heavy ions. Each particle interacts in a different way with matter:

- Photons: These particles are electrically neutral, they travel at the speed of light and can deeply penetrate matter. If a photon has a small energy, inferior to some MeV, it interacts with electrons by transferring a part of its energy (photoelectric and Compton effects). In this case, the electron is excited to another orbital layer or is freed from its atom and the photon continues its trajectory in the material. When the photon has an energy superior to some MeV, it is transformed into two particles (pair production): an electron and a positron [44].

- Neutrons: They are non-charged particles and they represent the most important part of natural particles at ground level that affect electronics. As they are neutral, they are very invasive and penetrate material deeply with small probabilities to interact. In fact, they deviate from their straight paths only to interact with a nucleus in the target material. Neutrons are not directly ionizing, but the resulting radiation from their collisions with nuclei is ionizing. They can be scattered or absorbed by the nucleus. Scattering can be elastic as the atom struck by the neutron leaves its crystalline structure due to the collision, or inelastic in case the neutron is captured by the atom and then released which results in the atom excitation. The absorption of the neutron by the struck atom results in the neutron mutation to another atomic element. In all cases, the neutron collision with the nucleus generates secondary ionizing particles which induce *indirect ionization* and are responsible for errors in electronic components [45][46].

- Charged particles: They can be protons, $\alpha-$particles or ions. They are highly energetic. Protons, for example, are unable to provoke direct ionizing effects for technologies superior to 90 nm but they result in nuclear reactions with material nuclei. For technologies of 90 nm and less, protons result in direct ionization. $\alpha-$particles and ions are responsible for single events by ionizing the material through the creation of electron-hole pairs. Packaging materials contain radioactive impurities which are strong alpha emitters. Alpha emitters are also present at wafer and interconnection levels. The proximity of alpha emitters to the silicon substrate makes alpha emission a concern even though the penetration depth of $\alpha-$particles is small [45][46].

1.5.3.2 Generalities on radiations effects on integrated circuits

Here is an overview of general radiation metrics. The *flux* is the number of incident particles per unit of surface and time and is expressed in $particles/cm^2s$. The integration of the flux over time gives the particle density, also called *fluence*, expressed in $particles/cm^2$. A particle transfers its energy to a material in case of interaction. *Linear energy transfer* (LET) is the amount of energy deposited by ionization and that goes through the material per unit path length. It is expressed in MeV cm^2/mg [47].

Two metrics allow to quantify the sensitivity of integrated circuits to ionizing particles: *the critical charge* and *the cross-section*. The critical charge fixes the threshold of collected charge that makes the target circuit sensitive to radiations. It is related to the threshold LET (LET_{th}). LET_{th}

is the minimal LET able to generate single events in an integrated circuit. LET_{th} is particular to the considered circuit and depends on the used technology. The cross section σ, measures the probability of occurrence of single events and represents the mean number of interactions of an incident particle per target entity over the received fluence. It is a function of LET and is expressed in $cm^2/device$ or cm^2/bit. The total sensitive surface of a circuit, also defined as the upper limit of σ for single events, corresponds to the saturation cross section σ_{sat} which can be determined from the characteristic curve that gives the cross section versus LET [48]. The typical shape of this curve is given in Fig. 1.22.

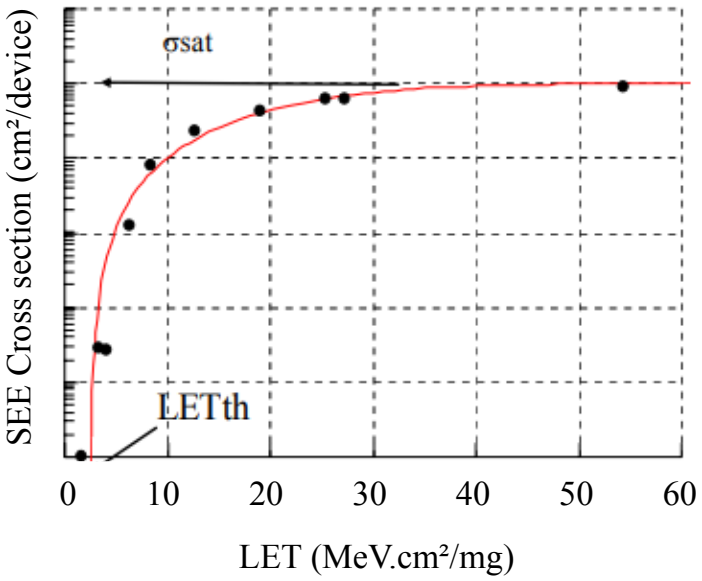

Figure 1.22: Typical curve of SEE cross section versus LET (adapted from [6])

The ideal curve that is used to to describe the cross sections of heavy-ion and proton-induced single events is given by the Weibull distribution when the values of LET_{th} and σ_{sat} are known and it follows (1.14) [49]:

$$\sigma = \sigma_{sat}\left[1 - e^{\left(-\frac{LET - LET_{th}}{W}\right)^{S}}\right] \tag{1.14}$$

where W is the width parameter of the rising portion of the curve and S is a dimensionless exponent that determines the shape of the curve.

1.6 Conclusion

Systems and devices reliability have evolved through the years due to transistor shrinking, in addition to the development of emerging technologies and new technological processes. The interest in reliability is further more justified for harsh environments characterized by exposure to high temperatures. The reliability of data storage devices at high temperatures is as important matter, as it is the basis for components accurate operations.

In order to provide an easier understanding of studies in the following chapters, the aim of this chapter was to give an overview of basic reliability notions for CMOS components exposed to high temperatures, with an emphasis on storage cells. Fundamental reliability notions were presented in the first part of this chapter, followed by a general study of temperature effects on MOSFET electric parameters. A second part was dedicated to NAND Flash memories, which are the non-volatile memories we chose to study in chapters 3 and 5. The structure and operation of these memories were detailed. The last part of this chapter was dedicated to SRAM, which are the volatile memories we chose to study in chapter 4. Similarly, the structure and operation of these memories were given, followed by details on radiation-related problems in integrated circuits.

Chapter 2

State-of-the-art of reliability monitoring techniques

For NAND Flash memories, data retention and cycling endurance are two important metrics that determine the reliability level. They are introduced in this chapter and followed by an overview of the state-of-the-art of monitoring techniques for retention and cycling endurance improvement. For 6T-SRAM, the radiation effects are presented with an emphasis on soft errors caused by Single Event Upsets (SEU). Finally, state-of-the-art techniques for SEU resilience are explained and a special design of a radiation-hardened storage cell is detailed.

2.1 Reliability monitoring in NAND Flash memories

2.1.1 Reliability fundamentals and metrics

The understanding of memory cells failure mechanisms is an important step for the estimation of NAND Flash level of reliability. NAND Flash memories are subjected to failures caused by oxide defects. Disturb mechanisms are also sources of data errors in NAND Flash memories. To measure the reliability of such memories, two metrics are chosen by manufacturers: endurance and data retention. Understanding the interconnection between these two metrics is mandatory to meet performance requirements and ensure systems reliability.

2.1.1.1 Effect of tunnel oxide properties on reliability

In NAND Flash memories, the *Fowler-Nordheim* tunneling mechanism, responsible for electrical charge injection and extraction from the FG, involves the use of high voltages and large electric fields across the gate oxide. The use of this same mechanism for both program and erase operations accentuates the degradation of the tunnel oxide of such memories [33].

The tunnel oxide is likely to loose its insulating characteristics with the accumulation of program and erase operations. Consecutive electron-tunneling operations result in molecular structure damages in the tunnel oxide. Atomic bonds are broken and oxide traps are generated with the capacity of electrons or holes trapping. Carriers trapping may lead to the increase of potential barriers and may affect the possibility to generate sufficient tunnel currents during program and erase operations. Trapped electrons lead to a proportional threshold shift ΔV_t that symmetrically increases V_t for both programmed and erased states. For an increasing number of program and erase operations, ΔV_t increases leading potentially to a read failure [7][34] (Fig. 2.1). When this

occurs, memory controller marks the involved blocks as bad and they are replaced by less worn out reserve blocks.

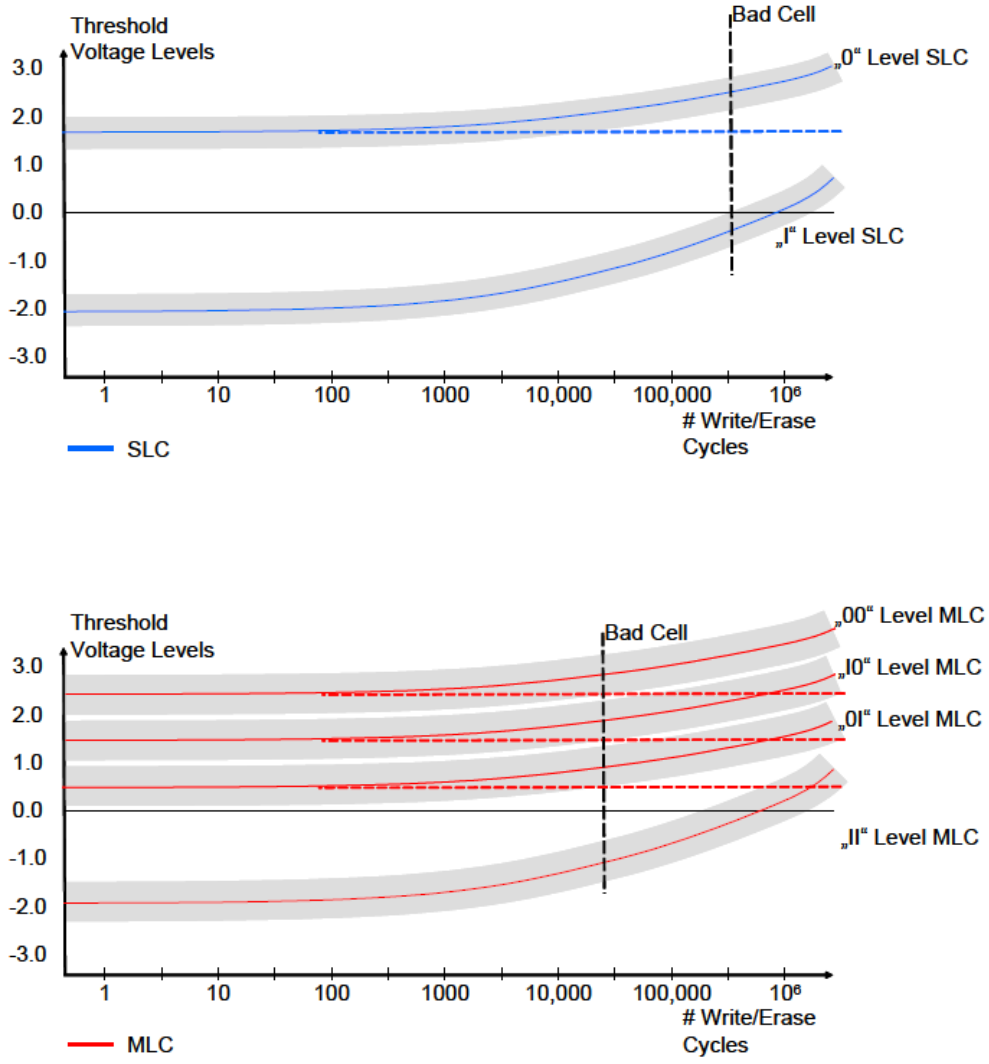

Figure 2.1: Threshold voltage variations with cumulative program and erase cycles for SLC and MLC Flash memories [4]

Carrier trapping can also lead to the increase of leakage current of OFF-state FG transistors that retain data. In fact, the oxide traps create a path suitable for the leakage of charges from the FG to the substrate. As a result, this can lead to stored information loss. This phenomenon is called Stress-Induced Leakage Current (SILC). SILC increases with the accumulation of program and erase operations and the decreasing thickness of the tunnel oxide [50]. SILC was identified as a limiting factor for the scaling of Flash memories tunnel oxide [51]. The structural changes can

eventually lead to the tunnel oxide total loss of the insulating feature called oxide breakdown (Fig. 2.2).

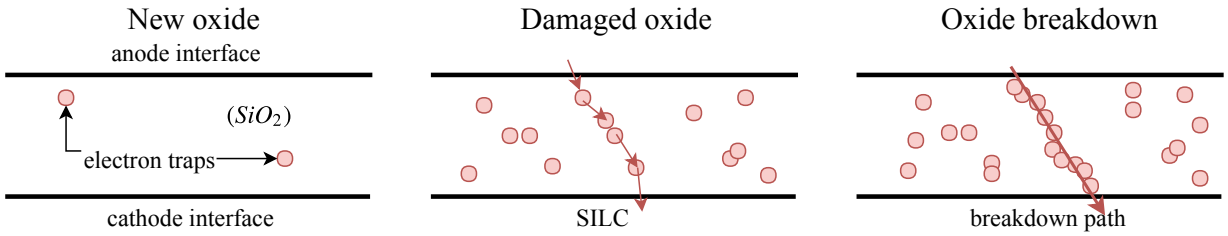

Figure 2.2: Stress induced leakage current and resulting oxide breakdown [7]

2.1.1.2 Memory architecture related disturbs

Due to NAND Flash memories architecture and array organization, an operation applied to a particular cell has an influence on the electrical potential of the floating gate of other neighboring cells. This influence is caused by the high density and capacitive coupling existing between memory cells. In addition, due to area efficiency constraints, voltage nodes are shared by different FG transistors which causes reliability issues called disturbs. The resulting disturbs are read, program and erase disturbs.

2.1.1.2.1 Read disturbs

Read disturbs result from the repetitive and consecutive reads operated without intermediate erase operations. Neighboring cells belonging to the same block, but not the same WL, as the read page are operated in ON state via the application of $V_{pass,R}$ independently from their programmed or erased state. This relatively high voltage along with the repetition of the $V_{pass,R}$ sequences may result in charge gain and partial programming of the neighboring cells. The V_t of these cells increases and shifts by a positive ΔV_t. This can lead to read errors that occur when the shift results in an increase of V_t above the read reference voltage. The probability of read disturb errors is higher for MLC than SLC because of the narrower V_t distribution window for each logic state. Read disturb errors are not permanent as erase and reprogram operations can restore the correct value in the erroneous read cell [8][52][53].

Fig. 2.3 shows the cells that may be affected by read disturbs in a NAND Flash block. Fig. 2.4 shows the effect of read disturbs on the threshold voltage distribution of an MLC.

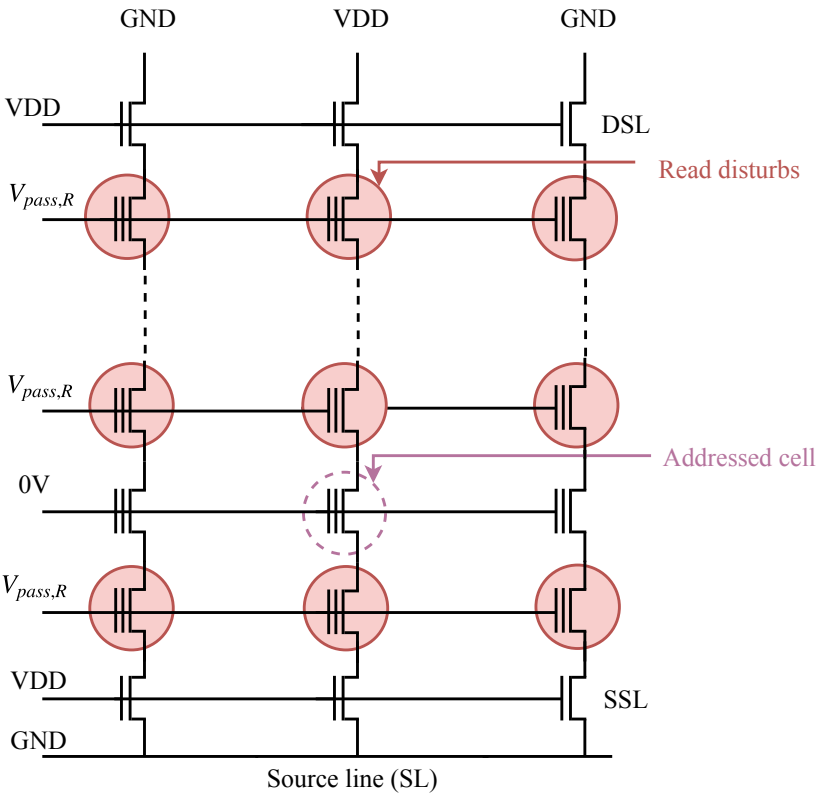

Figure 2.3: Potentially disturbed cells by a read operation in a NAND Flash block [3][8]

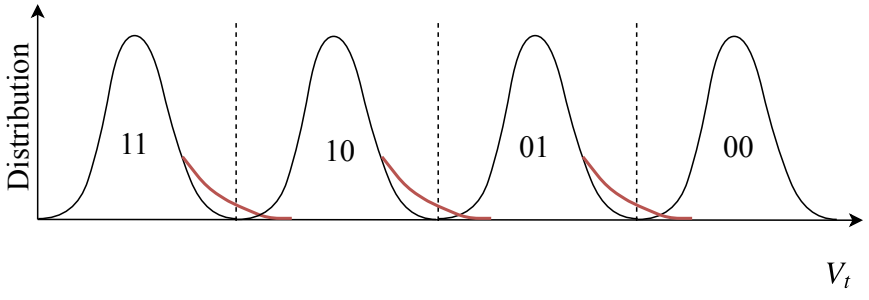

Figure 2.4: Impact of read disturbs on the threshold voltage distribution of an MLC [3]

2.1.1.2.2 Pass and program disturbs

Program operations result in two types of disturbs: the pass disturb and program disturb. During a program operation, cells belonging to the same grounded BL as the cell to be programmed have their CG subjected to $V_{pass,P}$, a relatively high positive voltage. As for read disturb, this voltage, combined with the grounded BL, enhance the electric field across the tunnel oxide and can result in an undesired charge transfer. This is called pass disturb. Program disturb affects *tunneling-inhibited* cells belonging to the same WL as the cell to be programmed. Soft-programming can occur in these cells even in the presence of program inhibit that boosts the channel potential as explained in section 1.4.2.2. The nature of such disturbs, soft-programming, is the same as in the case of read disturbs. The difference comes from the higher applied field and the higher involved voltages with respect to a read operation [8][52][53].

Fig. 2.5 shows the cells involved in pass and program disturbs among a NAND Flash block. Fig 2.6 shows the effect of program disturbs on the threshold voltage distribution of an MLC.

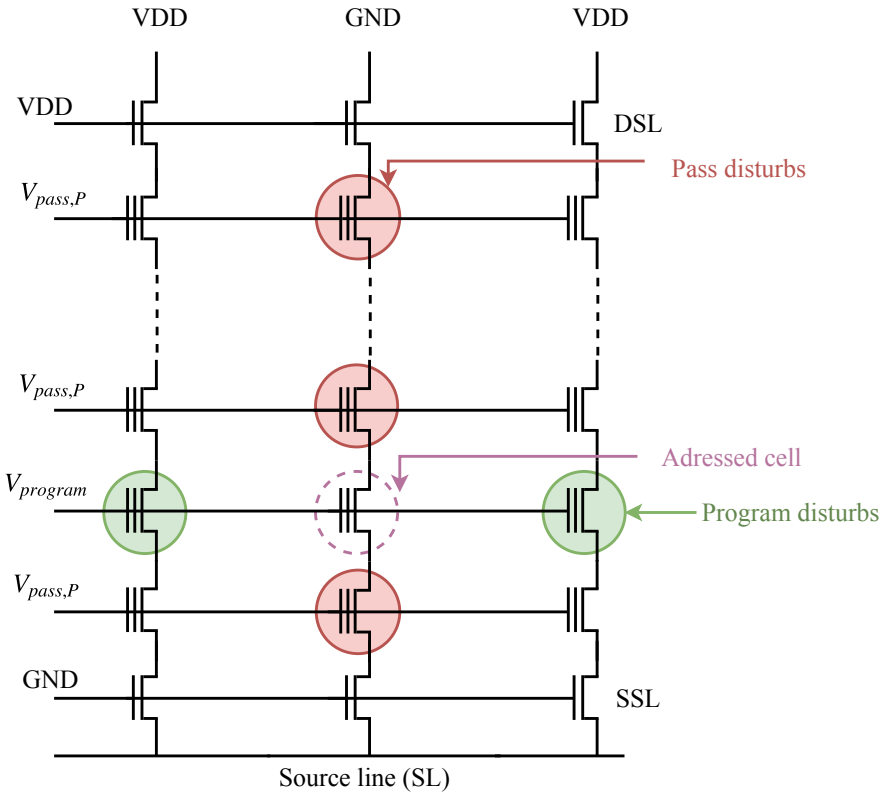

Figure 2.5: Pass and program disturbs in a NAND Flash block [3][8]

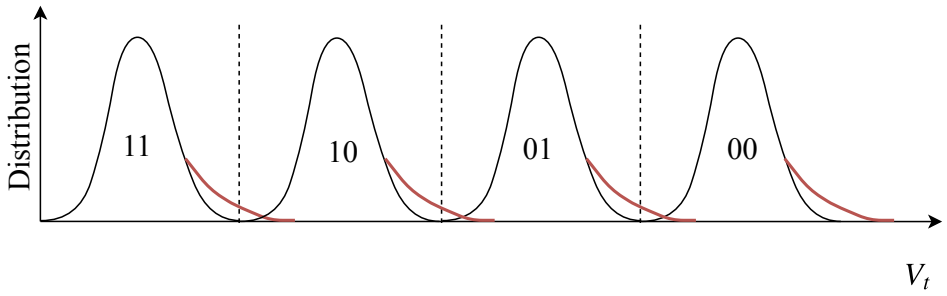

Figure 2.6: Impact of program disturbs on the threshold voltage distribution of an MLC [3]

2.1.1.2.3 Erase disturbs

Erase operations are done at a block level and there is no relevant block to block interference. However, as for program operations, a *self-boosted inhibit* mechanism appears in unselected blocks for the erase operation. A capacitive coupling occurs in cells, from blocks not intended to be erased, and together with the charged shared p-well, these cells see their WLs boosted. The tunneling of charges out of the unselected blocks is then prevented. This Self-Boosted Erase Inhibit (SBEI) can however suffer from an insufficient boosting of the WLs and results in a soft erase of programmed and unselected blocks for the erase operation. This is called erase disturbs which, unlike read and program disturbs, are of lower importance in NAND Flash memories [3].

Other than erase disturbs, NAND Flash memories can suffer from erase failures due to partial and incomplete erasure in selected cells. As explained in section 2.1.1.1 and due to the accumulation of program and erase cycles, the tunnel oxide quality is degraded. Charges trapping in the tunnel layer results in a positive shift of V_t of the erased state (Fig. 2.7). This can be interpreted as a programmed state of the NAND Flash cell when a read operation is performed.

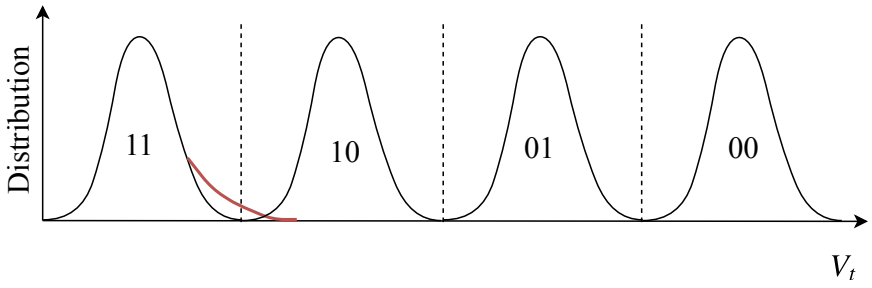

Figure 2.7: Impact of erase errors on the threshold voltage distribution of an MLC [3]

2.1.1.3 Endurance

In the case of a NAND Flash memory, endurance is a metric that describes the maximum number of cumulative program/erase (P/E) cycles that a given block can withstand before leading to a failure. Flash memory endurance is affected by the same failure mechanisms responsible for erase errors. In fact, operating a memory under high voltages introduces material defects in the tunnel oxide layer which are cumulative over time and which limit the endurance of a given block [3][4][8].

The maximum number of P/E cycles that can be withstood by the NAND blocks depends on the NAND Flash memory type. MLC NAND Flash blocks have less endurance than SLC ones. For example, SLC Flash memories of Cypress are able to accumulate up to $1,000,000$ P/E cycles while MLC Flash memories of the same manufacturer are able to withstand up to $100,000$ P/E cycles [16]. Another example are 24 nm SLC and MLC NAND Flash memories used in Swissbit solid state drives and that can respectively withstand $100,000$ and $3,000$ P/E cycles [4].

The nature of data to be stored can be correlated to the target Flash drive capacity in order to improve the endurance. The redistribution and equalization of erase cycles over the blocks can be considered and done through *wear leveling*. *Wear leveling* consists in evenly spreading P/E cycles over the whole Flash device. This prevents from having hot spots characterized by a high number of P/E cycles leading to a premature failure of the device. It is done at software level by the Flash File System (FFS).

2.1.1.4 Data retention

The ability to maintain the programmed data and provide it correctly on demand is a critical feature for NAND Flash memories. It is called data retention the maximum period of time during which data can be correctly stored and retrieved. Different parameters affect data retention including temperature, radiations, cumulative P/E cycles and cycling interval time [3][4][8]. Data retention can be given in the case of absence of cycling or for a specified number of cumulative P/E cycles. In fact, data retention and endurance are correlated metrics and there is a measurable relationship between the two parameters [4][16]. Material damage accumulation in tunnel oxide of NAND Flash cells caused by P/E cycles leads to SILC and potential loss of the stored information. As a consequence, failure to properly detect the initial programmed content of a cell leads to retention errors. Fig. 2.8 shows the threshold voltage distribution of an MLC suffering from retention errors.

Here are some examples of data retention of Flash memories. Uncycled Swissbit 24 nm SLC and MLC Flash memories of [4] provide a maximum data retention of 10 years at 40 °C. After $100,000$ P/E cycles, data retention of Swissbit SLC is reduced to 1 year at 40 °C, and for Swissbit MLC, data is retained for a maximum 1 year after $3,000$ P/E cycles at 40 °C. This is summarized in Table 2.1.

Another example is Cypress SLC and MirrorBit MLC Flash memories of [16] that are designed to provide a maximum data retention of 20 years at an average storage temperature of 55 °C. After

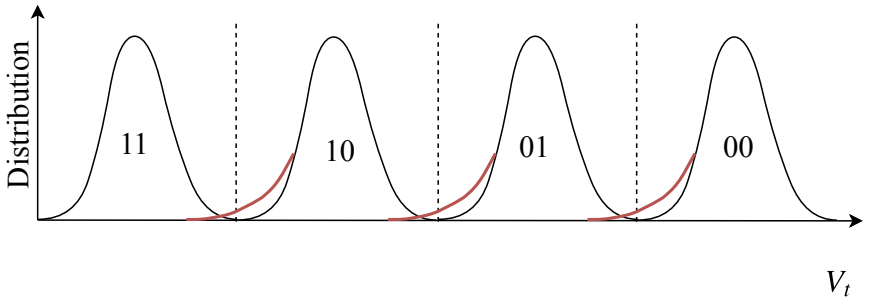

Figure 2.8: The threshold voltage distribution of an MLC with retention errors [3]

Cumulative erase cycles per sector	SLC typical data retention time post-cycling	MLC typical data retention time post-cycling
1 program	10 years	10 years
1,000 cycles	10 years	3 years
3,000 cycles	10 years	1 year
30,000 cycles	4 years	-
100,000 cycles	1 year	-

Table 2.1: Data retention and endurance relationship for Swissbit SLC and MLC Flash memories at 40 °C [4]

100,000 P/E cycles, data retention for Cypress SLC Flash memories is reduced to 1 year, while for MirrorBit MLC Flash, data is retained for a maximum 1 year after 10,000 cycles at an average storage temperature of 55 °C [16]. The relation between data retention and endurance in MirrorBit MLC Flash of Cypress is shown in Table 2.2. Here, it is assumed that the cumulative P/E cycles are uniformly inter-spaced over the lifetime of the device and that the average temperature, estimated from the consecutive periods of exposure to the different temperatures, is equal to 55 °C. More rapid cycling reduces the time interval between erase operations and leads to a reduced retention time.

Cumulative erase cycles per sector	Typical data retention time post-cycling
1 program	20 years
1,000 cycles	10 years
10,000 cycles	1 year

Table 2.2: Cypress MirrorBit data retention and endurance relationship at an average storge temperature of 55 °C [16]

2.1.2 Reliability improvement techniques in NAND Flash

In this section, we present an overview of the existing techniques for reliability improvement in NAND Flash memories. These techniques aim to enhance data retention time and cycling endurance. The basics of error correction techniques are firstly explained, followed by details on reliability improvement methods operated via aging-aware algorithms. The adjustment of the read operation is also a solution to grant reliability in NAND Flash memories. It is explained in the following. Lastly, solutions based on data refresh at fixed and adjustable frequencies are given.

2.1.2.1 Error correction codes

2.1.2.1.1 Basics for error correction codes

In NAND Flash memories, tunnel oxide deterioration and disturb mechanisms can lead to random bit errors among the stored data. The use of Error Correction Codes (ECC) is an adequate solution to protect against data corruption. ECCs are constructed by adding redundant parity bits to the data bits that have to be protected. One category of ECCs is block codes that are characterized by the calculation of redundant bits for a fixed size data block. When a $k-$bit of user data is written in a Flash memory, an encoder generates parity bits that are added to user data bits to form an $n-$bit codeword. The decoder searches for errors in the received codeword using the concept of binary hamming distance where differences are calculated between the received codeword and the valid one. The errors are then corrected within the correction capacity of the ECC (number of bit errors that can be corrected in a codeword) and the valid codeword is recovered [9]. Fig. 2.9 shows the encoding and decoding scheme of an error correction block code. One example of a block codes used in Flash memory system controllers is the Bose-Chaudhuri-Hocquenghem (BCH) code.

It is able to correct multiple random bit errors and it is characterized by overhead efficiency and computational complexity.

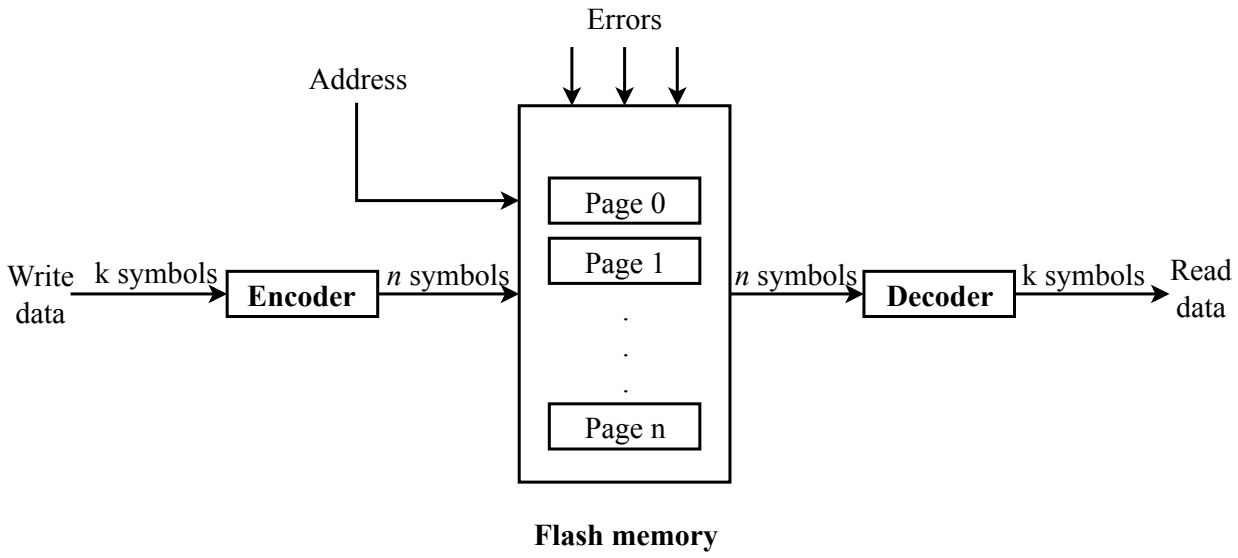

Figure 2.9: Encoding and decoding system of ECC in Flash memories [9]

2.1.2.1.2 Metrics of error correction codes

One important metric when considering NAND Flash reliability is the random bit error probability of a Flash cell called the Raw Bit Error Rate (RBER). The RBER is defined as the fraction of bits that contain incorrect data before applying an ECC [14] and is related to the metrics of section 1.1 by:

$$RBER = 1 - R(t) = 1 - e^{-\frac{t}{MTTF}} \tag{2.1}$$

where R(t) is the reliability function and MTTF the Mean-Time-To-Failure of the NAND Flash memory.

MLC are characterized by a higher RBER compared to SLC. Typical RBER in SLC NAND Flash is in the range of 10^{-11} to 10^{-9} errors per bit read. For MLC, typical RBER is in the range of 10^{-7} to 10^{-5} errors per bit read [7].

A $t-$bit BCH ECC able to correct up to t erroneous bits per codeword is able to detect, but not correct, at least $t+1$ bits per codeword [3]. The probability of corrupted data in the Flash memory is related to the number of errors per codeword that persist after the use of the ECC. For a $t-$bit BCH ECC, the probability of data corruption $P_{corrupt}$ is equal to the sum of probability of having all the error patterns of at least $t+1$ bits over the n bits of the codeword [7]:

$$P_{corrupt} = \sum_{i=t+1}^{n} \binom{n}{i} RBER^i (1 - RBER)^{n-i} \tag{2.2}$$

where RBER is the Raw Bit Error Rate representing the random bit error probability and where each bit can be corrupt or not independently from the others.

The Uncorrectable Bit Error Rate (UBER) corresponds to the ratio of data corruption probability over the number of bits in a codeword and can be written as follows [14]:

$$UBER = \frac{P_{corrupt}}{n} = \frac{1}{n} \sum_{i=t+1}^{n} \binom{n}{i} RBER^i (1 - RBER)^{n-i} \tag{2.3}$$

It should be noted that (2.2) can be approximated by the largest term which is the probability of having exactly $t + 1$ error bits. In this case, UBER is proportional to $RBER^{t+1}$.

2.1.2.1.3 Reliability improvement with error correction codes

The reliability of NAND Flash memories can be improved by an appropriate choice of the ECC error correction capacity. The choice of the ECC strength is a critical design issue that affects the development of NAND Flash storage systems. For example, Solid State Drives (SSD) which are used to replace Hard Disk Drives (HDD) may require a powerful error correction capability. ECC strength must guarantee a minimum UBER equal to the UBER of the HDD that the SSD replaces which is typically equal to 10^{-15}. With the expansion of MLC memories use in SSDs, providing multiple-bit corrections of stored data with ECCs is important in order to guarantee target reliability levels. Fig. 2.10 shows the relation between UBER and RBER for different strengths of a BCH ECC. A BCH ECC able to correct up to 7 errors in a codeword guarantees a good reliability for both SLC and MLC based SSDs.

A stronger ECC implies a higher decoding latency, storage and implementation overheads and power consumption. However, reliability metrics are degraded with aging due to the accumulation of P/E cycles, so the use of an ECC with a fixed and high correction capacity is not an optimal solution. One adequate solution for the reliability and performance trade-off consists in using the implementation of ECC in Flash controllers with a programmable correction capacity. This can adapt the ECC strength to the error rate changes with P/E cycles.

In [54][55], an algorithm that enables the selection at run time of an ECC correction capacity specific to each page of a NAND Flash memory is proposed. The selection is done using an online Flash BER calculated as a combination of an estimation model and real measurements deduced from the number of detected errors at a specific time. For the estimation model, this work presents an improvement that consists in considering both the number of P/E cycles and the retention age. Retention age is the storage time of data since it has been programmed or refreshed for the last time. Besides, the real measurement of the number of errors occurring during runtime gives the possibility of an accurate selection of the adequate error correction capacity. This algorithm can be

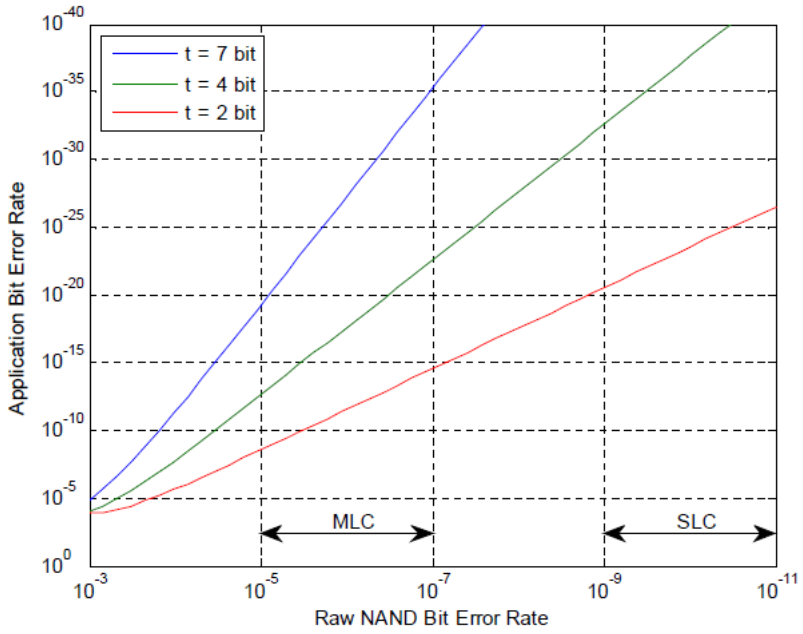

Figure 2.10: Relation between UBER and RBER for binary BCH error correction codes with different strenghts and over 512 bytes user data sectors [7]

adapted to the use of different types of ECC, including BCH codes, that are capable of providing information about the number of detected and corrected errors in a codeword.

In this study, The measured UBER takes into account the number of detected errors at a specific time, averaged by the page size. The page retention age is estimated as the difference between current time and a write timestamp that indicates the time of the last program or refresh operation of the considered data. The maximum retention time, that enables to retain the correct data and to sustain a selected UBER maximum value, is a function of the selected ECC correction capacity and can be pre-estimated offline. Each time the current retention time approaches the maximum retention time of the considered page, an alarm is issued and data needs to be refreshed.

2.1.2.2 Read operation optimization

One mitigation technique for NAND Flash reliability improvement consists in adapting the read reference voltage to the threshold voltage distribution shifts caused by the degradation mechanisms described in the previous sections. Different research groups showed interest in modeling program interference in MLC NAND Flash and proposed adapted mitigation techniques. Using the accurate estimation of threshold shift due to program interference, these research groups predicted the optimum read reference voltage between neighboring threshold voltage distribution states [10][56]. In fact for an MLC, if the threshold voltage distribution of two neighboring states are overlapping, the optimum read reference voltage that guarantees a minimal value for RBER is set at the crossing

point of the two threshold voltage distributions. A learning task allows the prediction of the program interference noise each 1,000 P/E cycles. The read reference voltage is based on the learned model: for a particular cell, if no neighboring cell is programmed, the default read reference voltage is used. Otherwise, the predicted read reference voltage is the sum of the default read reference voltage and the predicted threshold voltage shift.

As it can be seen in Fig. 2.11, the read reference voltage optimization method have been validated by RBER testing. The graph shows the reduction of RBER with this mitigation technique and for different P/E cycles. In fact, when keeping the same strength of the ECC, the P/E cycles lifetime of the Flash memory is improved. In the same way, keeping the same P/E cycles lifetime, a simpler ECC can be used to achieve the same RBER.

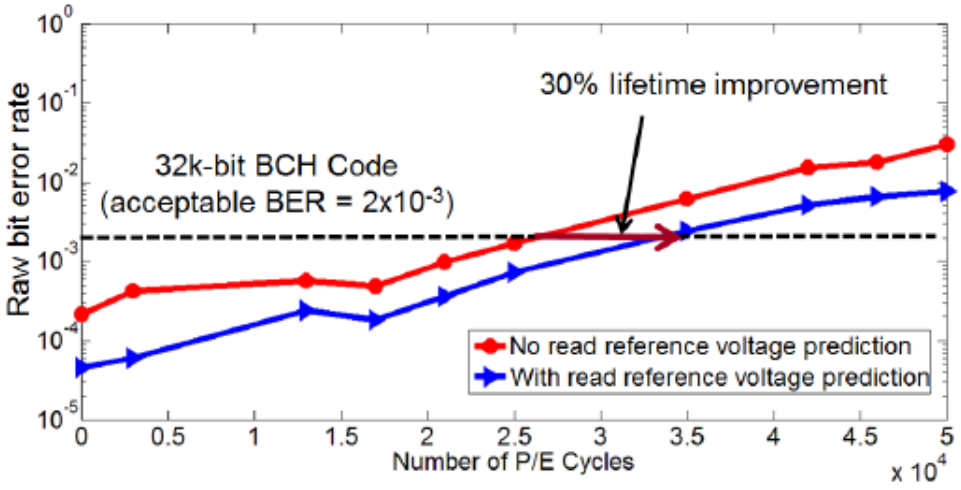

Figure 2.11: RBER with and without read reference voltage optimization [10]

The same research teams from Carnegie Mellon university and LSI corporation in San Jose also studied the effect of read reference voltage optimization technique on retention errors for 2Y-nm MLC NAND Flash memories [57]. Their major results are the following:

- They studied the distortion of the threshold voltage distribution at different P/E cycles and different retention ages, where retention age is defined as the period of time since a cell programming or refresh. They noted that:
  - Pages in the same block tend to have the same retention age, so the read reference optimization is considered block by block.
  - The threshold voltage distributions corresponding to the two most charged states of an MLC systematically shift to lower voltages faster than lower charged states.
  - The threshold voltage distributions become wider with a higher retention age.

- They characterized the optimum read reference voltage that allows achieving the lowest RBER. It shifts to lower values as retention age increases. Read reference voltage that separates the two most charged states of an MLC is more affected by retention age than the two least charged states.

- They estimated the optimum read reference voltage via *read-retry* technique. Here, the controller reads data out of a Flash memory step by step. It starts by a default read reference voltage. If the ECC successfully corrects data, the read step succeeds and the read voltage is the default one. Otherwise, the read reference voltage is changed to another value until a successful read operation. The *read-retry* technique is optimized by choosing a close to optimal starting read reference voltage and decreasing it by ΔV a minimal voltage step. The optimal read reference voltage is sensibly the same for all the pages of the same block.

This technique gives a minimal value of RBER compared to the cases where a read reference voltage optimized for a different retention age is used. In addition, the lifetime of a Flash memory can be extended when using the optimal read reference voltage. For a given RBER, the number of tolerable P/E cycles increases when the used read reference voltage is optimized.

2.1.2.3 Flash refresh and correction

One mitigation technique for NAND Flash reliability improvement consists in using the relation that exists between retention time and cycling endurance. In fact, when considering an ECC with a given strength, relaxing the constraint on retention time gives a considerable improvement in endurance and a much longer lifetime for the NAND Flash. In [58][13], this observation was used to propose a mitigation technique for retention errors based on periodic refresh and correction of errors. With the use of recent *wear leveling algorithms*, the following Flash Refresh and Correct (FCR) mechanisms were proposed:

- A *remapping-based FCR* mechanism that periodically corrects retention errors and remaps the affected pages to a different physical location. Flash memory lifetime is increased by 9 times when considering a fixed ECC strength and a fixed refresh period. On the other hand, considering a fixed P/E cycles lifetime requirement, the ECC strength can be highly simplified. This technique introduces unnecessary remap operations and considerable data movement with negative impact on read-intensive workloads. This can lead to faster wear-out and a reduced lifetime for Flash memories.

- An *in-place reprogramming-based FCR* mechanism that periodically corrects retention errors in-place without erasure and remapping. Re-injecting a precise amount of charge on floating gates, affected by charge leakage responsible for retention errors, enables the restoring of the originally stored values.

- An *hybrid FCR* mechanism based on the combination of both remapping and in-place reprogramming mechanisms. Using the ECC output, a count of the errors resulting from a

right shift in the threshold voltage distribution is done. So, the corresponding errors are all the errors other than retention ones and mostly program disturb errors. An in-place reprogramming is done in case this count is smaller than a certain threshold. Otherwise, a *remapping-based FCR* is applied to the whole considered block. This helps to avoid the accumulation of additional program disturb errors.

- An *adaptive rate FCR* mechanism that involves refresh by remapping or in-place reprogramming at an adaptable rate. The fixed rate refresh mechanism is pessimistic as refresh is not needed at the beginning of the Flash lifetime due to very low error rate. The refresh frequency of a block should increase over the block lifetime as the number of retention errors increases with P/E cycles number. The lifetime of a NAND Flash block can be divided into intervals depending on the number of P/E cycles. For each interval, a fixed refresh rate is used to meet the constraint on the acceptable RBER for the used ECC.

2.2 Radiation resilience techniques for SRAM

Scaled SRAM are particularly affected by the interaction of atmospheric particles at ground level (mainly neutrons). In this section, reliability issues of SRAM cells exposed to radiations are addressed with an explanation of soft errors and SEU. State-of-the-art resilience techniques against radiation effects are presented at the end of this section.

2.2.1 Radiation effects on SRAM

2.2.1.1 Cumulative dose and single events

Two types of radiation effects on integrated circuits can be distinguished: long-term effects called *cumulative dose* due to the accumulation of the dose and short-term effects called *single events* due to single particles ionization or secondary particles formation. Here, long-term effects caused by non-ionizing radiation such as Displacement Damage Dose (DDD) are not considered [59].

Cumulative dose is characterized by the *Total Ionizing Dose* (TID) which is a measure of the energy deposited by ionizing radiation in material per unit mass. In MOS technologies, it results from the accumulation, over time, of charges in the insulating oxide layers. This causes the drift of electric parameters as the shift of threshold voltage distributions and the increase of leakage currents. The effects of dose are cumulative so, they can result in permanent damage and total loss of the intended features of the considered device [6].

Single events occur when a single particle strikes the material deposing sufficient energy to cause an effect on the device. When ionizing particles penetrate the material, they can deposit charges in active areas of transistors which are then collected by electric fields. This results in the propagation of a current pulse that can provoke different Single Event Effects (SEE), that can be destructive (permanent) or non-destructive (transient). Here, non-destructive effects are studied. The most relevant ones for SRAM are Single Event Transient (SET) and Single Event Upset (SEU). SET

corresponds to a current spike provoked by an ionizing particle strike which propagates in a circuit. An SEU is a SET generating enough collected charge to be captured by a memory cell and resulting in its bit-flip. This represents a soft error and is detectable at the device output [6].

2.2.1.2 Soft errors in SRAM

In SRAM, soft errors are random and nonrecurring errors. They result from ionizing particle strike on a memory cell. Historically, $\alpha-$particles were the dominant source of soft errors at ground level because they result from the decay of impurities in packaging materials. This issue has been addressed by the use of highly purified materials. Nowadays, high energy neutrons from cosmic radiation constitute the main source of soft errors in atmospheric environment. A neutron is a neutral particle and its ionization is indirect as the nuclear reactions with the target material result in secondary charged particles. When a charged particle passes through the target material, this involves three main phenomenon [60]:

- *Charge deposition/generation:* Direct ionization of the target material results in charge deposition along the particle path. The same occurs in the case of neutrons via secondary particles resulting from nuclear reactions. With charge deposition along the particle track, energy is transmitted to electrons of struck atoms and secondary electrons are produced. This results in the creation of electron-hole pairs in the particle path.

- *Charge transport:* After charge deposition, the released carriers are transported in the target material and collected by elementary structures as p-n junctions. The transport occurs due to charge drift and diffusion. Charge drifts in regions with an applied electric field and is diffused in neutral regions.

- *Charge collection:* Charge transport results in charge collection and in parasitic current transients that induce disturbance in the target circuit. If the charge is collected in sensitive nodes of the target circuit, this can result in permanent or transient SEE depending on the intensity of the current transient and the number of sensitive nodes affected by the collected charge.

Fig. 2.12 shows the effect of a charged particle strike on a reverse-biased p-n junction which is highly sensitive to radiations. In fact, the deposited charge is collected very efficiently in these structures due to the strong electric field present in the depletion region [45].

In 6T-SRAM cells, the sensitive nodes to radiations are the drains of OFF-NMOS and OFF-PMOS transistors in the inverters pair. If the collected charge is comparable to the charge present in the node and exceeds a critical amount called critical charge Q_{crit}, then the corresponding voltage can be disturbed. The stored value can be flipped resulting in an SEU and the failures caused by these faults are called soft errors [61].

With technology scaling, the amount of collected charges caused by particles strike and the cross section for soft errors decrease. On the other hand, Q_{crit} decreases as a result of sensitive nodes

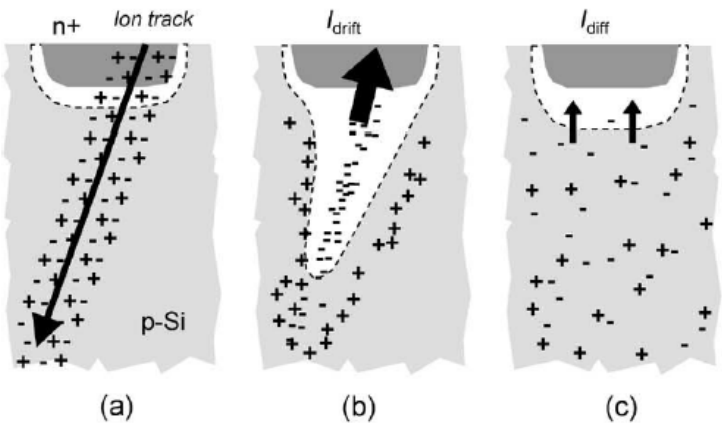

Figure 2.12: Generation (a), transport (b) and collection (c) of carriers after a charged particle strike in a reverse biased p-n junction [11]

capacitance and voltage decrease. This affects the Soft Error Rate (SER) as technology shrinks. SER is expressed in failure in time per Mbit (FIT/Mbit) with 1 FIT being defined as 1 event per 10^9 hours of use. For example, SER increases when considering the aircraft flight altitude compared to sea level as the neutron flux from cosmic rays increases [62].

2.2.2 Radiation effects resilience techniques

Different techniques have been adopted to improve systems and components resilience to radiation and to tackle the soft error problem. These solutions are divided into three groups: radiation shielding, components hardening and architectural hardening.

2.2.2.1 Radiation shielding

Radiation shielding consists in the use of physical barriers to protect from radiation effects and prevent ionizing particles from reaching sensitive cells in devices. The choice of materials and thicknesses of the physical barriers depends on the type of radiation and the energy involved. For example, die coating or underfill materials are chosen to shield $\alpha-$particles resulting from package contamination and lead bumps. This can also be used in the case of low energy neutrons and protons, but with less efficiency. However, this technique is inefficient in the case of high energy neutrons. For these particles, mitigation techniques involve process and design adaptation with architectural hardening [15][62].

2.2.2.2 Components hardening

Along with all the advantages related to the increase of integration density and performance

improvement, technology shrinking makes shielding increasingly difficult and induces additional radiation sensitivity problems. In this case components hardening is considered as a better alternative. It can be done on both technological or design levels. On the technological level, the hardening of the processes aims to make the components more resilient by using new materials and designing device layouts carefully. On the design level, new structures of basic electronic systems or functional blocs are proposed to insure less sensitivity to radiations.

Technology-based solutions consist mainly in chip manufacturing on epitaxial substrates, on insulating substrates and in the use of wide bandgap materials [63]:

- Epitaxial substrates are regular substrates with a lightly doped layer underneath. This reduces the sensitivity to latch-ups and improves the resilience to upsets.
- Silicon-on-insulator (SOI) technology is based on the replacement of a regular substrate by a layered silicon-insulator-silicon substrate. This induces lower parasitic capacitance due to junction isolation from bulk silicon. This technique also improves the resilience to latchups and upsets.
- Silicon-on-Sapphire (SOS) is part of SOI family and it is produced by growing a thin layer of silicon on a sapphire wafer. This type of wafer presents a high resilience to radiations as sapphire is an excellent electrical insulator and therefore prevents transient current caused by radiation from spreading to other elements on the chip. Its drawbacks are the complexity of process especially for scaled transistors and its higher cost.
- Wide bandgap substrates offer different advantages regarding high temperature and high frequency operations, optical properties and also radiation insensitivity. For example, Gallium Nitride (GaN) and Gallium Arsenide (GaAs) based components offer a high tolerance to dose effects and insensitivity to latchups but they present a high cost and high power consumption.

For design-based solutions, the attenuation of the transient impulsion resulting from a particle strike and the use of hardened structures are mainly considered. For critical transistors, the increase of channel length in order to reduce sensitivity to radiations is sometimes considered but it implies an increase in the transistor size. A first approximation of the critical charge able to flip the state of a storage cell is the product of threshold voltage by capacitance of sensitive nodes. Increasing the critical charge will reduce the effect of the transient impulsion resulting from a particle strike. This can be achieved by increasing the capacitance in sensitive nodes or the use of retro-action resistors to attenuate the current pulse [61].

The use of hardened storage structures is one of the most efficient solutions when transistor scaling is considered. In fact, 6T-SRAM cells have the smallest node capacitance and the weakest feedback which makes alpha particles and cosmic rays neutron flux impact a primary concern for these structures. The 6T-SRAM design can be hardened to radiation via the use of Dual Interlocked Cell (DICE) structure [12][61]. A DICE has twice as many transistors as in a 6T-SRAM and is used as storage CMOS cell featuring high resilience to upsets and soft errors. Its design is based

on spatial redundancy and feedback allowing the restoring of data in the case of a particle strike. The DICE design has four nodes (X1, X2, X3 and X4, Fig. 2.13) that are combined as two pairs of complementary logic states for data storage. During write/read operations, the complementary pairs are simultaneously accessed through four NMOS access transistors (NM5, NM6, NM7 and NM8).

In an ordinary 6T-SRAM cell, a particle strike that flips the state of an internal node can corrupt the stored value in the SRAM. The redundancy introduced in the DICE design provides a source of uncorrupted data after a particle strike. The uncorrupted section delivers the correct state and this way restores the feedback to recover the state of the affected node. It should be noted that the hardening feature of DICE cell is not dependent on optimal transistor sizing [64].

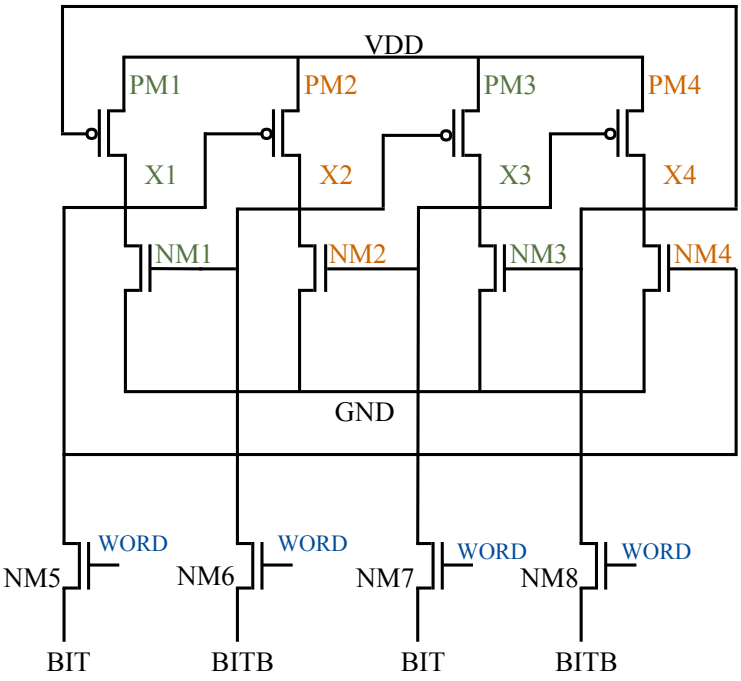

Figure 2.13: DICE cell structure [12]

2.2.2.3 System-level hardening

Another mitigation technique for radiations effects is system-level hardening. This makes components radiation tolerant and can also be combined with other mitigation techniques to reach the desired level of reliability. System-level hardening adopts various logical means, such as the use of error correction codes, redundant elements and watchdog mechanism [61][63]:

- ECC or Error Detection And Correction (EDAC) are used as mitigation techniques for radiation-induced errors. This technique requires periodic data scrubbing, i.e. a background task that periodically inspects memory for errors, to prevent radiation effects and errors accumulation. This technique makes the memory radiation-tolerant, depending on the nature of the ionizing particle and its energy, as long as an adequate strength is used for the ECC. A compromise should be found with the overhead introduced by a stronger ECC and how critical the application is.

- Redundancy is a widely used technique for SET and SEU mitigation. Redundancy can be either spatial or temporal. Spatial redundancy consists in replicating the material resources of the system such as processors and memories. Triple Modular Redundancy (TMR) is the most used and consists in the use of a vote among the results of three different replicas of the same module. Performance is not degraded with this solution but both cost and electrical consumption increase. Temporal redundancy consists in the sequential execution of the same task followed by a vote. This technique is very useful to mitigate SEU effects as it is sensitive to bit flips within the logic part of a circuit. This solution induces losses of performance without modification of the hardware part of the application.

- Watchdog mechanism can also be used to protect against radiation effects. It is a hardware or software surveillance mechanism that can be implemented in different levels of the system to verify that it receives a signal (i.e. the system is alive). If radiations cause the system to operate incorrectly and the watchdog timer does not receive a signal, a recover operation is then performed. This can be done through a hard reset. A last solution consists in the reboot of the faulty system and the periodic download of critical information in order to predict future errors. This technique is considered as a last resort to other radiation mitigation techniques.

2.3 Conclusion

In this chapter, we gave an overview of the state-of-the-art techniques for reliability improvement in NAND Flash and radiation resilience in SRAM. In NAND Flash memories, oxide failure and disturb mechanisms are responsible for data errors. Data retention and cycling endurance are used to measure the reliability of such memories. The existing monitoring techniques for NAND Flash memories aim the enhance these two metrics using the interconnection that exists between them. These techniques include the use of adequate error correction codes and an improvement method based on aging-aware algorithms. The optimization of the read operation was also presented as a reliability improvement technique. Finally, solutions aiming to adapt data refresh frequencies were detailed for NAND Flash memories.

For SRAM, reliability issues of cells exposed to radiation were presented with an emphasis on soft errors and Single Event Upsets (SEU). Radiation resilience techniques ranging from shielding to components and system-level hardening were addressed at the end of this chapter.

Chapter 3

Data refresh methodology in Flash-SSD based on Arrhenius Timer

The first part of our work concerns the retention of data stored in Flash memories. In this chapter, we introduce a data refresh methodology for NAND Flash memories used in SSDs. This refresh scheme is based on the integration over time of temperature effects on data retention capability, which are described by the Arrhenius law. A special module, we called *Arrhenius timer* or *A-timer*, is introduced in this chapter. It allows the reduction of the number of data refresh operations, and thus, the improvement of NAND Flash memories reliability. The first section concerns the retention errors rate in NAND Flash memories and the impact of temperature variations on it. Limits of other mitigation solutions based on refresh are then introduced. In the second section, the A-timer purpose and operation are explained. Its structure along with simulation results of the reduction rate of the refresh frequency are detailed. The last section is about the possible use of the A-timer with timestamps for warning triggering.

3.1 Endurance and retention issues in NAND Flash memories

NAND Flash-based SSDs offer an alternative to HDD. They are characterized by better performance and lower power consumption. In addition, the absence of moving parts makes them more durable and more resistant to shock and vibration. The price of SDDs is in constant decrease in order to meet HDD-comparable cost requirements. This is achieved by technology scaling and the use of multilevel cells in NAND Flash memories to decrease the cost per gigabyte. Unfortunately, this comes to the detriment of affecting the reliability of NAND Flash memories. The cumulative number of program and erase (P/E) cycles that can be withstood by flash memories, i.e. the cycling endurance, is decreased by one decade when switching from SLC to MLC in SSDs [16][65]. The same decrease by one order of magnitude in cycling endurance is noticed for each additional bit stored in an MLC [3][4]. In fact, higher data storage capacity results in faster wear-out. In addition, the maximum period of time during which information can be correctly stored, i.e. data retention time, in NAND Flash-based SSDs is also affected by technology shrinking and MLC use. Data retention capacity is highly sensitive to operating temperature variations and to the number of endured P/E cycles. As the number of P/E cycles increases, data retention capability is negatively affected [4][16]. This negative effect is further enhanced by the increase of temperature.

Due to retention and endurance mutual dependence, mitigation solutions aiming to improve one of these reliability aspects can be also used to optimize the other. In fact, the maximum number of

P/E cycles that can be withstood by a Flash memory is increased if the constraint on retention time is relaxed. This means that mechanisms allowing to cope with retention time reduction, caused by temperature increase for example, can be used to improve cycling endurance. Periodic check and refresh is an efficient solution to deal with data retention degradation and to extend the endurance of NAND Flash-based SSDs.

3.1.1 Retention error rate in NAND Flash

NAND Flash memories suffer from various types of errors. Some of them, resulting from loss of stored information, are categorized as retention errors. These errors are affected by memories operating conditions and the accumulation of program and erase cycles. The other types of errors are due to the particular architecture and array organization of Flash memories. Errors due to the influence of an operation applied to a particular cell on the data stored in a neighboring cell are categorized as disturb errors. These errors can be read disturb, program disturb or erase errors.

Previous studies aiming to characterize retention errors, read disturb, program disturb and erase errors in NAND Flash memories confirm the following points:

- In SLC NAND Flash, retention and disturb tests are compared in case of retention tests acceleration by an an annealing at the highest temperature of the specified operating range. The tests results show that retention errors are dominant and no disturb errors are detected [52].

- The characterization of NAND Flash reliability prior to use, that includes endurance cycling, data retention bakes and disturb testing, is more important and justified for MLC than SLC NAND Flash memories [53].

- In MLC NAND Flash, all types of errors (retention and disturb errors) are highly correlated to the number of P/E cycles. The different error rates increase exponentially with P/E cycles [13][66].

- In MLC NAND Flash, a significant error rate difference exists between the various types of errors. Retention errors for data with high retention age (time elapsed after programming and before testing a memory for retention errors) are the most dominant. They are followed by program disturb errors, read disturb errors and finally erase errors as it can be seen in Fig. 3.1 [13][56][57].

- In MLC NAND Flash, the retention error rate is highly dependent on the retention age. It increases linearly with the retention test time. For example, as it can be seen in Fig. 3.1, the 3-year retention error rate is almost three orders of magnitude higher than the 1-day retention error rate [13][57].

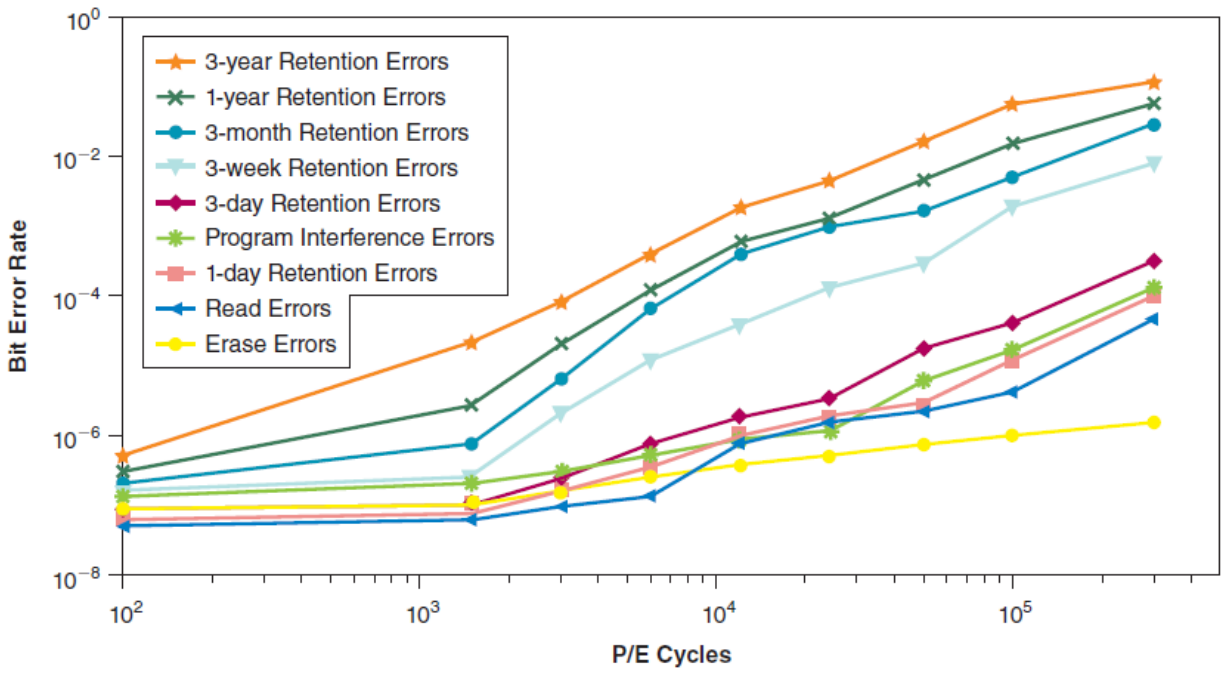

Figure 3.1: Error rates variation with P/E cycles for the different types of errors in 3X-nm MLC NAND Flash memories [13]

These different observations justify the choices that we made in the following study. In fact, we focus on retention errors in the case of NAND Flash memories as these errors are predominant over the other types of errors. Besides, retention errors are the ones that affect the most the bit error rate. Moreover, we chose to consider SSDs based on MLC NAND Flash since, in this type of memories, the retention problem in exacerbated. With the A-timer, we aim to cope with the retention time reduction that can be caused by temperature increase in the case of MLC NAND Flash-based SSDs.

3.1.2 Effect of temperature on retention capability

Fowler-Nordheim tunneling mechanism is responsible for charge injection and extraction from the floating gate in NAND Flash memories. As it involves the use of high voltages and large electric fields, the tunnel oxide is likely to loose its insulating properties over time and with the accumulation of P/E cycles. With carrier traps appearing in the tunnel oxide layer, leakage current probability increases for OFF-state floating gate transistors retaining data. High temperatures exacerbate the effect of defects in the tunnel oxide. This leads to the amplification of reliability problems related to trap-assisted tunneling and stress induced leakage current, especially when NAND Flash memories have endured a large number of P/E cycles. Thus, temperature has a negative impact on retention

capability in NAND Flash memories. Temperature is more impacting in the case of MLC than SLC NAND Flash memories as they present higher sensitivity to physical changes in the tunnel oxide layer.

High Temperature Data Retention (HTDR) test is used as an acceleration method to characterize NAND Flash memories and estimate their lifetime [67][68]. In HTDR tests, the Arrhenius model allows lifetime estimation through the Acceleration Factor (AF) that gives the relation between the retention times at a temperature of use and at a higher stress temperature. The AF is given by (3.1) [67][69][70][71]:

$$AF = \frac{\tau_{RET}(T_{use})}{\tau_{RET}(T_{stress})} = \exp\left[\frac{E_a}{K}\left(\frac{1}{T_{use}} - \frac{1}{T_{stress}}\right)\right] \tag{3.1}$$

where τ_{RET} is the retention time, T_{use} and T_{stress} the absolute temperatures of use and stress respectively, K the Boltzmann constant and E_a the activation energy of the different failure mechanisms responsible for the retention time reduction.

Under the effect of temperature, different failure mechanisms occur concurrently and contribute to data loss in NAND Flash memories. Each one is characterized by an activation energy E_a usually determined by empirical measurements. One of the failure mechanisms can be dominant over the others depending of the used technology or considered temperature range. The main failure mechanisms responsible for retention time reduction in NAND Flash memories are [72][73]:

- *Detrapping* mechanism caused by trapped electrons leaving their traps in tunnel oxide under the effect of thermal energy. This results in a threshold voltage shift in retention tests. This shift is exacerbated in the case of cycled NAND Flash cells as the number of traps existing in the tunnel oxide is higher and the probability of electrons tunneling through them during retention bake is also increased. The activation energy E_a of detrapping mechanism is reported to be between 1.1 eV and 1.2 eV.

- *Trap Assisted Tunneling (TAT)* mechanism is responsible for Stress Induced Leakage Current (SILC) as it consists in carriers tunneling through the oxide in two or more steps under the effect of thermal energy. This also results in a shift in the threshold voltage which is higher for cycled NAND Flash and for higher bake temperatures. In fact, with cycling, the distance between the traps is smaller as they are higher in number in the tunnel oxide. The activation energy E_a of SILC is reported to be equal to 0.3 eV or less.

Each of these failure mechanisms follows the Arrhenius law and contributes to the retention characteristic and lifetime estimation of the NAND Flash through its own acceleration factor and activation energy E_a. The higher the activation energy, the more sensitive the failure mechanism to temperature increase [69]. The general retention characteristic of a NAND Flash is based on the use of an apparent activation energy E_{aa} which is derived from the different activation energies of the involved failure mechanisms [72][73].

Without loss of generality, for our data refresh methodology in SSDs, we consider both of the failure mechanisms separately to estimate the gain in refresh frequency over a chosen range of temperature. The endurance improvement based on our data refresh methodology is mostly suitable for systems that have a short downtime. This is the case of enterprise-class SSDs in data centers which have very short periods of power-off [74]. In addition, enterprise-class SSDs are characterized by better performances and more demanding reliability requirements compared to client-class SSDs. In fact, enterprise-class SSDs should ensure a power-off data retention time of 3 months at 40 °C. This requirement is used to illustrate in Fig. 3.2 the Flash memories retention time versus temperature for both detrapping and SILC mechanisms in an enterprise-class SSD following the JEDEC requirements [68][75].

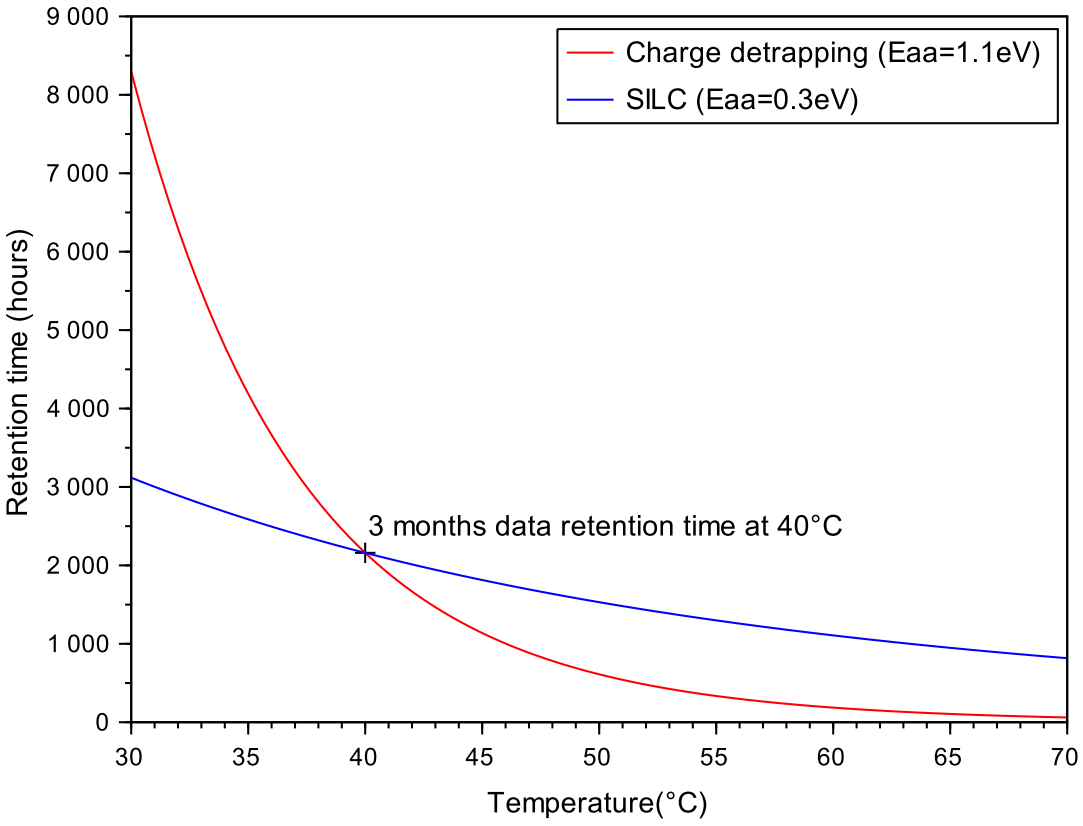

Figure 3.2: Flash memory retention time versus temperature for an enterprise-class SSD respecting the JEDEC requirements (JESD218B.01)

3.1.3 Limitations of conventional refresh schemes

The mutual dependence between retention time and cycling endurance has already been investigated to improve reliability in NAND Flash memories. The relaxation of the constraint on retention time by refresh, combined with the consideration of an adequate ECC, have been used to extend the lifetime of NAND Flash. Nevertheless, the already existing solutions present some limitations.

The refresh schemes in remapping-based Flash Correct and Refresh (FCR) and in-place reprogramming FCR [58][13] allow the improvement of MLC NAND Flash lifetime when the refresh frequency and ECC strength are fixed. These solutions present the limit of having a fixed refresh frequency which is excessively pessimistic. In fact, a fixed refresh frequency does not consider the evolution of the bit error rate over the lifetime of the NAND Flash. In addition, it results in a negative impact on response latency and energy consumption. These solutions also do not adapt to the variable workloads. For example, remapping-based FCR is not adapted to read-intensive workloads as it introduces additional remap operations. On the other hand, in-place reprogramming FCR scheme results in the accumulation of program disturb errors. These limits are addressed by the use of an hybrid FCR scheme [13][58] where both remapping and in place-reprogramming are used depending on errors count. This adapts more to the workload type and avoids additional program disturb errors appearance. Moreover, limitations caused by fixed refresh frequency are addressed by the use of adaptive rate FCR schemes [13][58] that adapt the frequency of refresh to the number of accumulated P/E cycles. This is done by dividing each blocks lifetime to intervals where a fixed refresh rate is used in correlation with the strength of the used ECC. In these refresh schemes, even though the refresh frequency is adaptable, it does not follow the temperature variations that highly influence retention in NAND Flash memories. Our idea is to extend the adaptable refresh rate to the temperature variations through the use of the Arrhenius timer.

Our study is also orthogonal to other mitigation techniques that only aim to reduce the number of retention errors or cope with the limited cycling endurance of NAND Flash memories. To reduce the number of retention errors, read reference voltage optimization technique adapts the read reference voltage to the retention age, i.e. the period of time separating the read operation from the last program or refresh operation [57]. This solution must be aware of the data retention age and does not take into account the temperature variations. Arrhenius timer needs to be only aware of the temperature variations in order to integrate their effect on retention time and reduce the number of retention errors by adapted refresh. In addition, wear leveling allows to cope with the limited cycling endurance by the redistribution and equalization of erase cycles over the blocks depending on their wear-out degree [3][7][4]. This improves the lifetime of a Flash memory without considering its retention. Arrhenius timer integrates both aspects to improve the reliability of NAND Flash-based SSDs.

3.2 Arrhenius timer purpose

The refresh scheme proposed in this chapter, and based on the use of the A-timer, allows the reduction of the number of data refresh operations in NAND Flash memories through the integration

of temperature effect on their retention time.

When temperature variations highly affect data retention time in NAND Flash memories, the use of a worst case refresh frequency can be avoided by:

- Regularly checking the operating temperature.
- Calculating the time integral of the temperature impact on data retention time.
- If the time integral reaches a threshold value that indicates a potential retention hazard, then a warning is triggered and a refresh operation is executed.

This is suitable in the case of enterprise-class SSDs used in data centers as downtime is very short which allows efficient temperature tracking even when the power supply is off. In addition, many SSDs today contain at least one temperature sensor, so the time integral of the temperature impact can be computed by the SSD controller if it is regularly interrupted to execute the following operations:

- Check the output of the temperature sensor.
- Select the temperature acceleration factor from an pre-computed table.
- Add the selected value to the content of a status register.
- If the difference between the current value of the status register and its value during the last warning triggering or program operation exceeds a certain threshold that indicates a potential retention hazard, then a warning is triggered.

Due to the small number of instructions in this monitoring procedure and the relative infrequency of its occurrence, the performance overhead is small. But with the increase of the number of Flash chips in SSDs, a temperature gradient may be induced and the use of several temperature sensors may be needed. Performance and power overheads quickly rise with the multiplication of the different operations for each temperature sensor.

This can be avoided by the use of a hardware module, the A-timer, able to achieve the operations described above. It allows to mainly compute the time integral of temperature impact on retention and to trigger alerts against potential data retention hazards. In the following, we detail the structure of the A-timer.

3.3 Arrhenius timer structure and operations

The proposed A-timer implementation as shown in Fig. 3.3 is composed by:

- A digital temperature sensor to detect the operating temperature.
- A multiplexer and registers to select the adequate increment value corresponding to the sensed operating temperature.

- A ripple-carry adder to add the increment value to the last status register value.
- A status register updated by the output of the adder at a low frequency.
- A low frequency clock generator for the proper functioning of the status register as the temperature variations rate is lower than the system clock frequency.

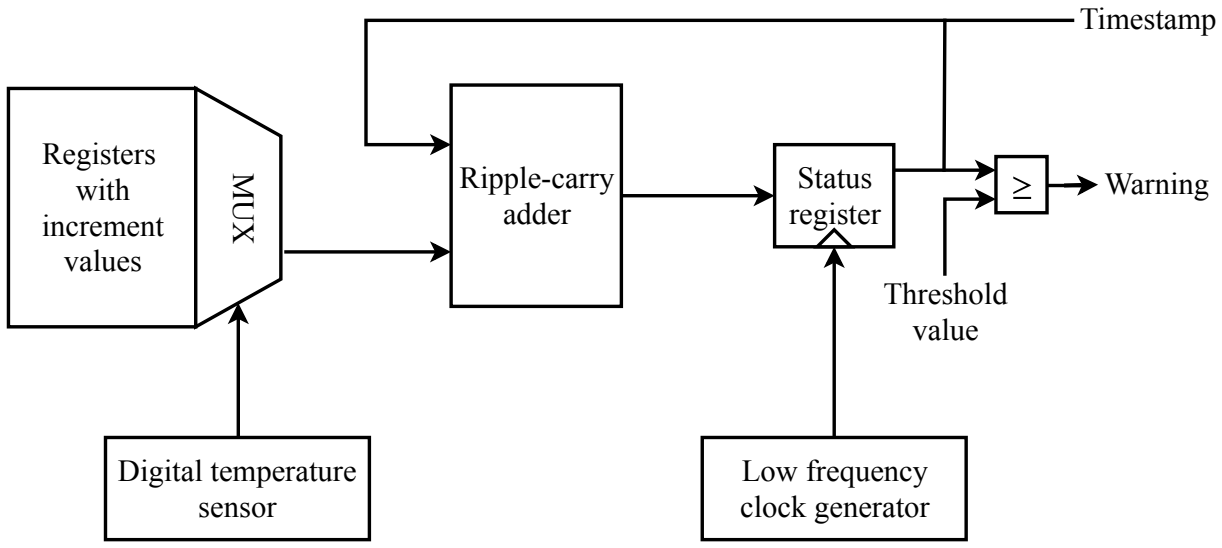

Figure 3.3: Block diagram of A-timer implementation

An A-time warning is issued when the output value of the status register reaches a threshold value. The status register is not reset at each warning in order to use its output value as a timestamp. Thus, the threshold value of the A-time is incremented by Δth_{inc} each time a warning is triggered. This Δth_{inc} represents the difference to the next status register state for which an alert has to be issued.

Here, the value of the status register is considered as a timestamp that integrates the effect of temperature variations. In each memory block and upon an A-timer warning, the remaining retention time is evaluated based on the difference between (a) the A-time state, which is the status register current value, and (b) a timestamp provided by the A-timer itself, which is the value of the status register during the last warning triggering or program operation. Data needs to be refreshed if the difference is above a certain threshold that can be also dependent on the number of endured P/E cycles.

The output value of the A-timer can be considered as a temperature-weighted timestamp. This timestamp can be seen as a metadata assigned to each memory block, along with the number of

endured P/E cycles, validity flags, logical address mapping information, etc. Timestamps have already been proposed to evaluate data retention age and used to estimate the RBER in Flash memories for mitigation techniques based on the use of ECC with adaptable strength [54][55].

In the A-timer implementation, the status register is updated at a constant rate by the adder and the frequency generator, but with variable increment values in order to approximate the temperature impact on the guaranteed data retention time as it can be illustrated by Fig. 3.4. The dashed curves represent the approximated data retention time that can be guaranteed by the A-timer depending on the considered temperature interval. They must be maintained below the solid curve representing the retention time given by the Arrhenius law in order to avoid data retention hazards.

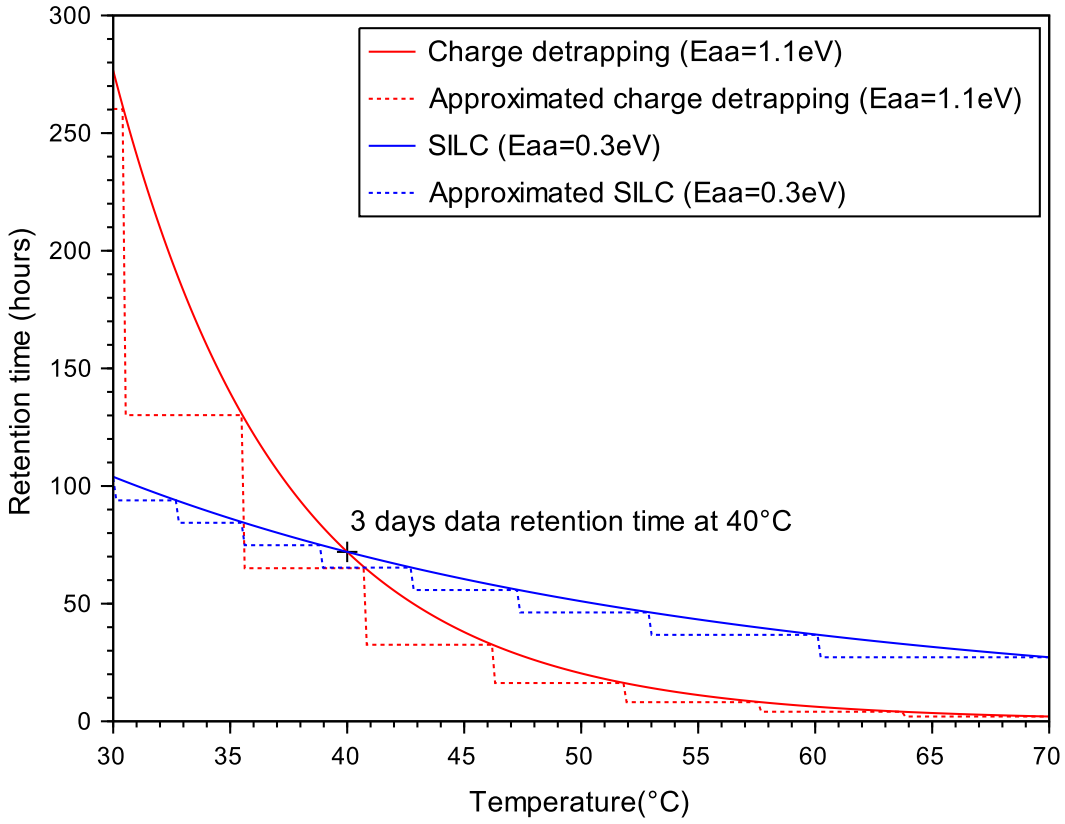

Figure 3.4: Arrhenius curves approximation that can be achieved by Arrhenius timer

The A-timer module is adapted to NAND Flash-based enterprise-class SSD. In data centers, ambient temperatures stay around 21 °C to 23 °C, and inside SSDs, temperature can reach the

30 °C-70 °C interval. Thus, for the illustration of Arrhenius curves approximation with the use of the A-timer, we chose the 30 °C-70 °C temperature interval as it can be seen in Fig. 3.4. Here, a pessimistic fixed refresh frequency would be the one that guarantees the minimal retention time for the interval 30 °C-70 °C, i.e. the retention time for 70 °C. Instead, with the use of the A-timer, the 30 °C-70 °C temperature interval is divided into sub-intervals. For each sub-interval is associated a constant approximated data retention time τ_{ret_app}, that corresponds to the minimal retention time for the sub-interval.

We chose to consider data retention curves approximation for both detrapping and SILC mechanisms. As the activation energies are different, the effect of temperature is not as notable for the two mechanisms in the considered 30 °C-70 °C temperature interval. The red dashed curve represents the data retention time approximation corresponding to detrapping. It is obtained by dividing the temperature interval into 8 sub-intervals in order to have τ_{ret_app} divided by 2 at each transition from one interval to the following. Here, τ_{ret_app} can be deduced from the formula below (3.2) that involves the increment period $\tau_{CLK} = \frac{1}{f_{CLK}}$, the increment value of the status register $inc(T)$ and Δth_{inc} the increment value of the A-time threshold value:

$$\tau_{ret\_app}(T) = \tau_{CLK} \frac{\Delta th_{inc}}{inc(T)} \tag{3.2}$$

When considering the Arrhenius law for detrapping mechanism, retention time is reduced by a factor of 136 when increasing temperature from 30 °C to 70 °C. The 8 sub-intervals are then characterized by increment values equal to 2^i with $0 \leq i \leq 7$, 2^7 being assigned to the sub-interval that starts at 70 °C.

For SILC mechanism, a similar approximation method is considered. The blue dashed curve represents the temperature interval divided into 8 sub-intervals over which τ_{ret_app} is constant. When going from a sub-interval to the following, τ_{ret_app} is reduced by 35% of the guaranteed retention time at 70 °C in order to fill the 280% gap between retention times at 30 °C and 70 °C in 8 steps. Here, the increment values can also be deduced from (3.2).

For the detrapping mechanism, the steps of data retention time approximation are logarithmically-spaced whether they are linearly-spaced in the case of SILC. This choice is justified by the higher activation energy of detrapping mechanism that results in a higher effect of temperature on retention time compared to SILC.

In Fig. 3.4, the Arrhenius solid curves have been scaled (from Fig. 3.2) in order to have a guaranteed data retention time of 3 days at 40 °C which allows to double the number of P/E cycles as compared to the situation of Fig. 3.2. This can be useful in the case of enterprise-class SSDs in data centers as power-off periods are very short and equal to a maximum of few hours per year [74].

The use of the A-timer presents different advantages over a processor-based solution. One of the advantages is the smaller 2-input NAND-equivalent gate count in A-timer compared to a low power processor. A similar advantage of A-timer over low power process-based solution is expected for

power consumption. A possible option for the A-timer is employing the deep sleep mode between consecutive temperature measurements. In this case, the proportion of time during which the timer is active is minimal. When using A-timer for SSD in data centers, the small shortage time can be overcome by the use of small backup batteries. Another approach consists in saving the state of the status register of the A-timer in a non-volatile memory before the power-off period, in addition to considering the worst case temperature effect on retention time during the power-off period.

3.4 Estimated refresh frequency reductions

Here, we consider different temperature distributions between 30 °C and 70 °C in order to estimate the refresh frequency reductions induced by the use of the A-timer. Symmetrical and asymmetrical temperature distributions were considered through uniform, normal and Gamma distributions. The standard deviations of the normal and Gamma distributions were set to 3.3 °C and 6.6 °C which respectively represent 1/12 and 1/6 of the whole temperature range. Fig. 3.5 represents the probability density distributions considered for the estimation of refresh frequency reductions with a standard deviation equal to 3.3 °C. The dashed curves are mirror images of the solid curves with respect to a vertical middle axis at 50 °C.

If we consider the approximations of the guaranteed data retention time given by the dashed curves in Fig. 3.4, then the mean of the approximated data retention time $\tau_{ret\_app\_mean}$ can be calculated as follows:

$$\tau_{ret\_app\_mean} = \sum_{i=0}^{7} \frac{CDF(T_{i+1}) - CDF(T_i)}{CDF(70\,°\text{C}) - CDF(30\,°\text{C})} \cdot \tau_{ret\_app}(T_i) \tag{3.3}$$

where:

- CDF is the cumulative distribution function of the temperature distributions in Fig. 3.5.
- $CDF(T_{i+1}) - CDF(T_i)$ is the probability that the operating temperature is in the interval $T_i - T_{i+1}$.
- $\tau_{ret\_app}(T_i)$ is the approximated data retention time in the interval $T_i - T_{i+1}$.
- T_i represent the temperature values where dashed curves in the approximation of Fig. 3.4 give the same retention time values as the Arrhenius solid curves.

For each distribution, we estimate the refresh frequency reduction with respect to the worst case refresh frequency and to the ideal refresh frequency. The results for both standard deviations 3.3 °C and 6.6 °C for the different distributions are reported in Table 3.1. The improvement with respect to the worst case refresh frequency is estimated as the ratio between $\tau_{ret\_app\_mean}$ and the guaranteed data retention time at the highest operating temperature of the range, i.e. 70 °C. The comparison with respect to the ideal refresh frequency is given by the ratio of $\tau_{ret\_app\_mean}$ and the expected data retention time for the ideal Arrhenius curve approximation.

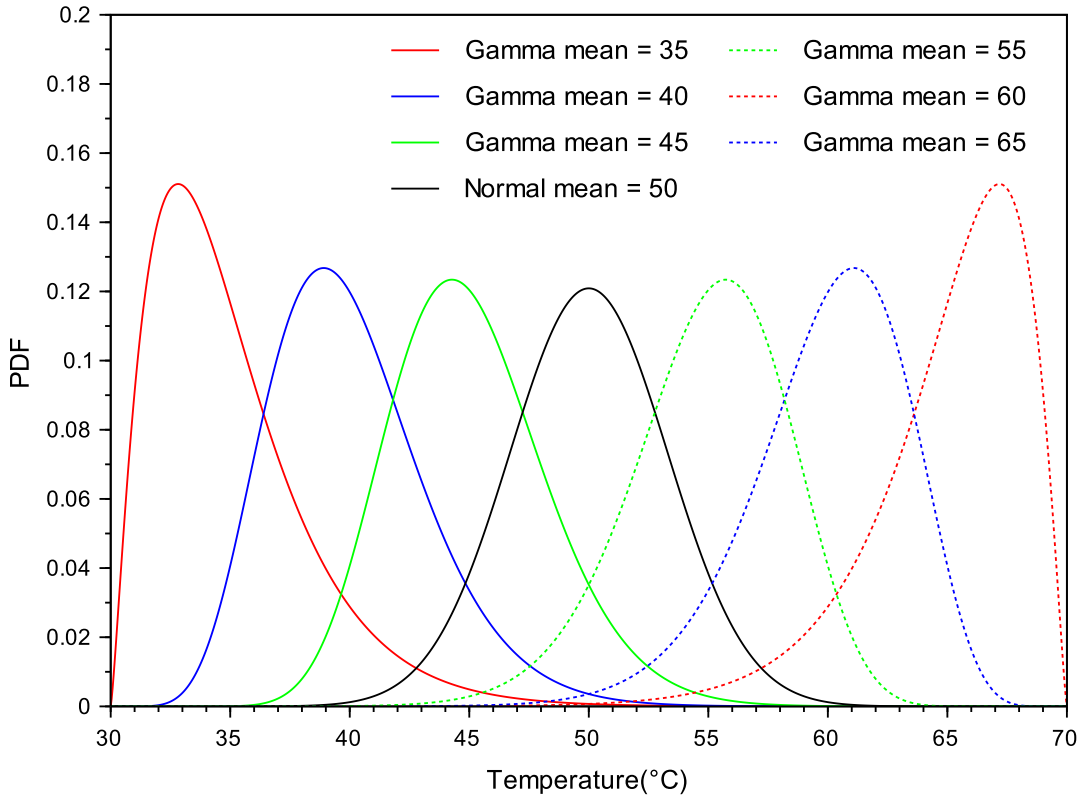

Figure 3.5: Probability density functions of the considered normal and Gamma distributions of the operating temperature with a standard deviation of 3.3 °C

Distribution	Mean value (°C)	Standard deviation (°C)	Refresh frequency reduction compared to			
			Worst case refresh frequency (at 70 °C)		Ideal refresh frequency	
			1.1eV	0.3eV	1.1eV	0.3eV
Gamma	35	6.6	62.71	3.01	0.72	0.92
Gamma	35	3.3	52.53	2.92	0.70	0.91
Gamma	40	6.6	33.49	2.47	0.71	0.90
Gamma	40	3.3	27.81	2.42	0.72	0.90
Gamma	45	6.6	18.10	2.06	0.72	0.90
Gamma	45	3.3	14.62	2.01	0.72	0.90
Uniform	50	11.5	18.40	1.83	0.73	0.89
Normal	50	6.6	10.13	1.71	0.72	0.88
Normal	50	3.3	7.88	1.68	0.72	0.89
Gamma	55	6.6	5.85	1.44	0.73	0.88
Gamma	55	3.3	4.34	1.43	0.72	0.89
Gamma	60	6.6	3.37	1.21	0.73	0.87
Gamma	60	3.3	2.42	1.15	0.73	0.84
Gamma	65	6.6	2.07	1.08	0.74	0.91
Gamma	65	3.3	1.37	1.02	0.72	0.87

Table 3.1: Estimated refresh frequency reductions between 30 °C and 70 °C

We can see that the refresh frequency reduction with respect to the worst case refresh frequency at 70 °C is sensitive to the activation energy E_a of the considered failure mechanism and the shape of the operating temperature distribution. Here, refresh frequency is mostly reduced for the higher value of $E_a = 1.1$ eV compared to $E_a = 0.3$ eV and for distributions biased towards the lower values of temperature in the operating range 30 °C-70 °C. The lowest refresh frequency reduction ratio is observed for the Gamma distribution centered at 65 °C. The Highest refresh frequency reduction ratio is observed for the Gamma distribution centered at 35 °C and it is equal to 62.7x for $E_a = 1.1$ eV and to 3x for $E_a = 0.3$ eV. With a wider operating temperature range, the ratio of refresh frequency reduction is expected to be higher.

For a fixed value of the activation energy E_a and if we consider different distributions with the same mean value, we have a higher ratio of refresh frequency reduction for larger standard deviations. This is due to the higher time spent at lower temperatures and thus the higher $\tau_{ret\_app\_mean}$. For example, the uniform temperature distribution gives a higher value of refresh frequency reduction ratio compared to the two normal distributions with standard deviations of 3.3 °C and 6.6 °C.

The ratios of refresh frequency reduction with respect to the ideal refresh frequencies from Arrhenius curves have values around 0.71x for $E_a = 1.1$ eV and 0.87x for $E_a = 0.3$ eV. This means that the ideal refresh frequencies are smaller than the mean of the approximated refresh frequency resulting from the use of the A-timer. The values of the approximated refresh frequency are closer to the ideal ones in the case of $E_a = 0.3$ eV. This can be explained by the fact that for this activation energy value, a lower variation rate of the Arrhenius curve is observed compared to the case of the activation energy $E_a = 1.1$ eV. This results in smaller approximation errors and a ratio closer to 1.

3.5 Warning triggering with A-timer and timestamps

The advantage of an A-timer based refresh scheme over a processor-based solution for enterprise-class SSD, is the ability to use one A-timer for all flash memories covered by a temperature sensor. The use of a single A-timer is possible when a timestamp is associated to each Flash memory block. Each time a block is erased, the corresponding timestamp needs to be updated with the program operation time of the first page of the block. In addition, the output of the A-timer status register can be used as a timestamp. When a warning is issued, the state of the A-timer is compared to the timestamps of each memory block. If the difference is higher than a certain threshold, the valid pages of the corresponding block are refreshed. This technique has the advantage of allowing the refreshing of the blocks that have the same functional update frequency. The block pages will have the tendency to update and be invalidated simultaneously which allows to reduce the overall number of blocks that may require refresh.

Since one A-timer is used to verify the timestamps of different blocks, the time period between the warnings τ_{warn} of the timer should always be smaller than the approximated retention time of all the blocks. It is also necessary that τ_{warn} is smaller than $\tau_{ret\_app}/2$ to avoid refreshing all the memory blocks each time the A-timer warning is triggered. If we take $\tau_{warn} = \tau_{ret\_app}/2$, then we

can see on Fig. 3.6 the moments when two blocks B1 and B2, arbitrary programmed between t(i) and t(i+1), have to be refreshed. In this case, the two blocks need to be refreshed at t(i+2). Thus, B1 is refreshed shortly before data retention time τ_{ret_app} is exhausted, as it was first programmed shortly after t(i). This is not the case of B2 which is refreshed shortly after $\tau_{ret\_app}/2$, as it was first programmed shortly before t(i+1). This induces an underestimation of data retention time by a factor between 0 and $\tau_{warn} = \tau_{ret\_app}/2$ in addition to the Arrhenius curve approximation errors. The use of one A-timer for all the memory blocks comes at the price of this underestimation. This error must be taken into account only for the first refresh operation after a program operation of a block. In addition, this underestimation can be reduced by scaling τ_{warn} to τ_{ret_app}/n with $n > 2$. This induces an underestimation between 0 and τ_{ret_app}/n. In this case, it is necessary to find the value of n that offers the best performance, power consumption, and endurance trade-offs.

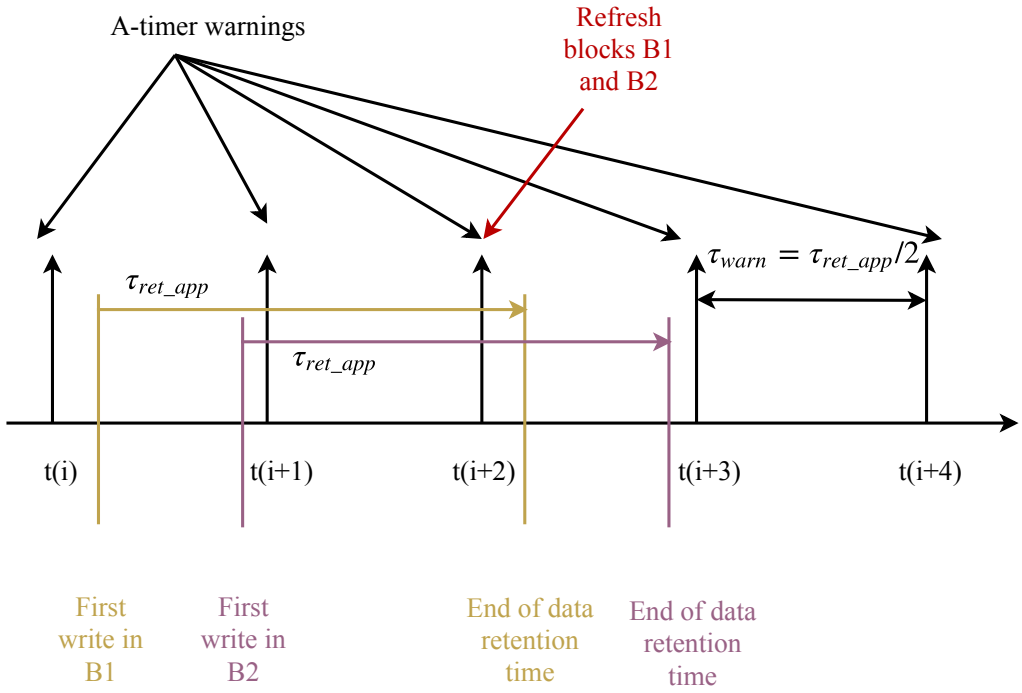

Figure 3.6: Refresh operations with the use of A-timer warning period τ_{warn} and the approximated data retention time $\tau_{ret\_app} = 2\tau_{warn}$ considered at a constant operating temperature for two arbitrary data blocks B1 and B2

The A-timer can be used to adapt to the number of P/E cycles endured by memory blocks. This is done in addition to wear leveling approaches at SSDs that take into account the hotness of data. In fact, cold data characterized by small functional update frequencies are stored at blocks that can endure the largest number of P/E and shortest retention time in order to compensate for

the lack of hotness of data with the refresh operations. Blocks with different values of endured P/E cycles can have different guaranteed data retention times, so different approximated data retention times τ_{ret_app}. In this case, the relation between τ_{ret_app} and τ_{warn} can be defined as $\tau_{ret\_app} = floor(\tau_{ret\_app}/\tau_{warn})\tau_{warn}$, τ_{ret_app} is then approximated to the smallest multiple of τ_{warn}. This way, some blocks are refreshed on the second A-timer warning after the first program operation, some other blocks are refreshed on the third A-timer warning and so on. This allows the adaptation of the refresh technique as the refresh frequency is modulated by the number of endured P/E of each memory block.

To evaluate the storage overhead of the A-timer timestamps, we consider a 256 GB SSD with 32 raw flash chips. A Flash chip contains 2^{13} blocks, each block with 2^7 pages of 8 KB each. The SSD has a total of 32 x $2^{13} = 2^{18}$ blocks, so the use of A-timer will involve 2^{18} timestamps for the whole SSD. We consider that the A-timer has a unique temperature sensor and is used to measure the operating temperature every second for three year. We consider the red dashed curve in Fig. 3.4, so the maximum incremental value of the A-timer status register is equal to 2^7. In this case, the status register needs to have a maximum of 32 bits.

The timestamps can be shorter than the status register states as the least significant bits of the register can be neglected. In this case, the measurement error corresponding to the estimation of data retention time is negligible. By considering a timestamp of 28 bits, the storage overhead is compared to a block of 1 MB (1024 x 1024 x 2^8). So the whole SSD will require 28 x 2^{18} bits for timestamps storage and can be stored in a single memory block. The timestamps of the A-timer can be uploaded into the RAM of the SSD controller with the ability to save this content in a non-volatile memory when the power is switched off. Besides, the performance overhead of checking 1 MB of timestamps is negligible. Following the scheme of Fig. 3.6, the timestamps need to be checked once every 36 hours, if the τ_{ret_app} is equal to 3 days.

3.6 Conclusion

In NAND Flash memories, reliability can be improved by investigating the mutual dependence between retention time and cycling endurance. Thus, memories lifetime can be improved by relaxing the constraint on retention time and using an adequate ECC. This can be achieved by periodic refresh schemes. When important operating temperature variations are involved, a fixed frequency for refreshes can be excessively pessimistic as it must be adapted to the highest considered temperature. A special module, A-timer, is proposed to reduce the number of refresh operations in MLC NAND Flash memories, through the integration of temperature variations effect on their retention time. The Arrhenius law is considered for the retention time temperature dependence for detrapping and Stress Induced Leakage Current (SILC) failure mechanisms, both responsible for data retention hazards. A-timer allows to efficiently approximate the impact of temperature changes for MLC NAND Flash used in data centers SSDs. For asymmetric temperature distributions over 30 °C-70 °C temperature interval, the refresh frequencies are reduced by up to 63x and 3x for detrapping and SILC mechanisms, respectively. A single A-timer can also be used for refresh operations in all the

blocks of the considered memory. In this case, warning should be triggered by using the states of the A-timer as timestamps.

Chapter 4

Improvement of the tolerated raw bit error rate in NAND Flash SSDs

In this chapter, we introduce a technique to improve the tolerated Raw Bit Error Rate (RBER) in enterprise Flash-based SSDs used in data centers. The tolerated RBER is estimated in relation to JEDEC-compliant Uncorrectable Bit Error Rate (UBER) values. RBER in the studied memories is dominated by retention errors rate. The proposed technique relies on this observation. Using a strong ECC to achieve the target UBER can be avoided by statistically estimating the left retention time in each memory page and by employing an adequate refresh scheme based on it. The left retention time is the maximum reliable remaining data storage time that guarantees the respect of the target UBER. In the first section of this chapter, details on a detection technique of retention errors among other types of errors in MLC NAND Flash-based SSDs are given. This is followed by the statistical background for left retention time and retention RBER estimation. In the second section, a refresh scheme based on embedded statistics for the left retention time estimation is explained. The third section gives the simulation results of improvement ratios of the maximum tolerated RBER. The resulting refresh probability and average time between refresh operations of the statistical scheme are then detailed. The last section gives an alternative based on a linear approximation of the tolerated retention RBER for left retention time estimation. Similarly, average time between refreshes in addition to access time reduction are given compared to a systematic refresh scheme.

4.1 Retention errors detection and rate estimation

As explained in section 3.1, NAND Flash-based SSDs are used as an alternative to HDDs due to their lower power consumption and better performance. With the use of MLC NAND Flash, SSD price is decreasing, but this comes with reliability metrics degradation as cycling endurance and retention. In this chapter, we consider enterprise-class SSDs based on MLC NAND Flash that are characterized by more demanding reliability constraints. In fact, JEDEC standard recommendations fix a maximum value of UBER equal to 10^{-16} for enterprise-class SSDs against 10^{-15} for client-class SSDs [68]. A recent large-scale study of MLC Flash-based SSDs in the data centers of Facebook showed that UBER can significantly exceed the standard values fixed by JEDEC. These SSDs presented UBER values between 10^{-11} and 10^{-9} [76]. The use of a strong ECC improves UBER but comes with high latency and also implementation and power overheads. A better handling of the RBER can improve reliability in MLC NAND Flash memories even with an ECC of a smaller

fixed strength. Periodic refreshes with remapping or in-place reprogramming (section 3.1) can be used to contain the values of RBER. These periodic refresh schemes are generally based on worst case scenarios. They can lead to unnecessary latency and power overheads, in addition to faster wearout related to excessive P/E cycling [13][58].

As explained in section 3.1, retention errors are dominant over the other error types in MLC NAND Flash memories, especially when longer storage periods are considered. In addition to be dominated by retention errors rate, RBER in MLC NAND Flash memories can be much higher in some blocks compared to others. This is a result of process variations and uneven wearout [77]. In data centers, RBER can also be much higher in small ratios of MLC NAND Flash-based SSDs than in others. These small ratios highly contribute to the overall UBER degradation [76].

In this section, a method to distinguish retention errors from other error types in MLC NAND Flash is presented. The statistical background of retention RBER estimation is then detailed, as it highly contributes to the overall RBER of these memories.

4.1.1 Retention errors detection in MLC NAND Flash

MLC NAND Flash memories are affected by different types of errors that can be divided into two groups: disturb errors and retention errors. Disturb errors result from the interference between neighboring cells in the NAND array and they include read disturb, program disturb and erase errors. Retention errors affect the ability of a memory cell to keep the stored value unchanged during a required period of time. Retention age is defined as the data storage time since the last program of refresh operation. Retention errors result from the progressive loss of information due to leakage current and the accumlation of material damages in the tunnel oxide [56][57][58]. Here, the special logic state encoding of an MLC Flash memory of Fig. 4.1 is considered. Most significant bit (MSB) and least significant bit (LSB) correspond respectively to the first and second bits starting from the left in the considered logic states encoding. Only MSB or LSB changes in the case of transitions occurring between neighboring logic states. Thus, retention errors can be easily distinguished from the other types of errors as it will be detailed in the following.

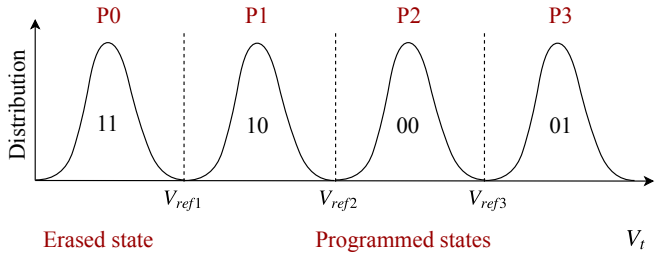

Figure 4.1: A logical state encoding of an MLC NAND Flash and the corresponding threshold voltage distribution

Retention errors result from a negative shift of the threshold voltage distribution to the lower values, which causes the crossing of read reference voltages. During read access, a logic state can be sensed as a lower neighboring logic state. Disturb errors result from a positive shift of the threshold voltage distribution to the higher values which also causes the crossing of read reference voltages. During read access, a logic state can be mistaken for a higher neighboring logic state. The effects of the different errors on the threshold voltage distribution are illustrated in Fig. 4.2.

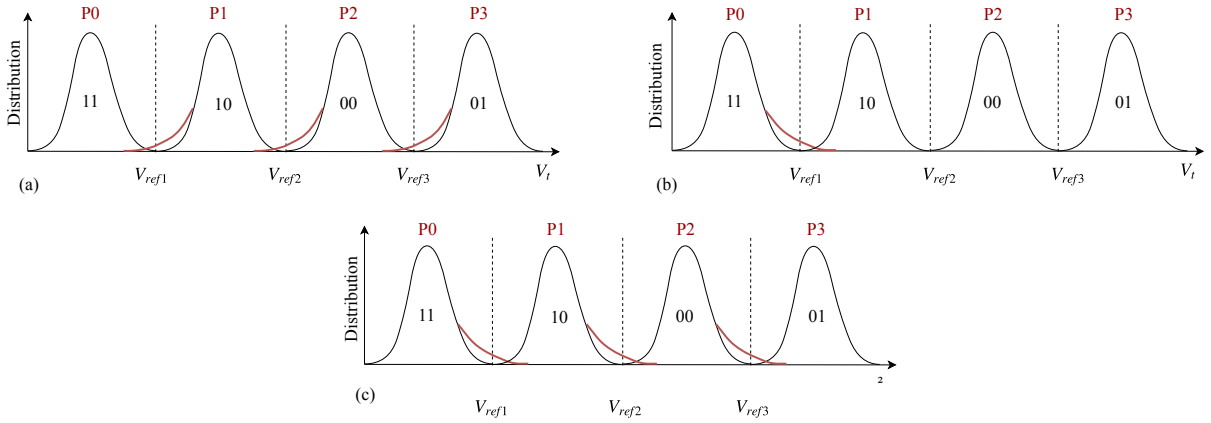

Figure 4.2: Effects of retention errors (a) erase errors (b) and read and write disturb errors (c) on the threshold voltage distribution of an MLC NAND Flash memory

Program disturb errors are characterized by the second highest rate in MLC NAND Flash after retention errors with high values of retention age. They occur due to capacitive coupling and affect wordline neighboring cells after that a program operation takes place in a cell of the same wordline. If we consider a fully programmed block of a Flash memory and if pages of the block are not refreshed, program disturb errors rate won't increase over the time [56][57][58].

Read disturb errors are the consequence of repetitive read operations on the same string of a NAND memory block without intermediate erase operations. When an erase operation is performed on a memory block, erase errors occur in cells failing to reset to the erased state. Read disturb and erase errors are both characterized by low occurrence in MLC NAND Flash memories when compared to retention errors with high values of retention age [56][57][58]. In the following, we consider that the RBER of MLC NAND Flash memories is dominated by retention errors rate, which is the only error rate to increase with retention age.

Due to these differences, retention errors can be detected and distinguished from the other error types with the help of an ECC. In a Flash memory, read operations are followed by an error correction step where data is checked. Erroneous bits are then identified and corrected to their intended initial value. With the help of the ECC, the polarity of each error is assessed, which is the difference between the erroneous bit value and its corrected value. In the case of the encoding of

the MLC NAND Flash of Fig. 4.1, we additionally assume that disturb and retention errors only involve transitions between neighboring logic states. In fact, neighboring logic states transitions have a higher probability of occurrence in MLC NAND Flash [13]. Retention errors can then be distinguished via the error fingerprints of Table 4.1. In this table, the first line corresponds to fingerprints of a retention error causing a shift from P2 to P1 state in an MLC NAND Flash with the encoding of Fig. 4.1. This also corresponds to the fingerprint of a retention error in an SLC cell characterized by the storing of a single bit. Line 2 corresponds to the fingerprint of a retention error causing the shift from P1 to P0 state and line 3 to the fingerprint of a retention error causing the shift from P3 to P2. Thus, all the error retention cases are covered in Table 4.1.

Considered bit of the MLC Flash cell	Read value	Corrected value	Value of the companion bit (LSB/MSB)
MSB	1	0	1 or 0
LSB	1	0	1
LSB	0	1	0

Table 4.1: Retention errors fingerprints of an MLC NAND Flash with the encoding of Fig.4.1

4.1.2 Statistical background for retention RBER estimation

As explained in section 1.1, failure rate is an important metric to consider when studying the reliability of electronic components and devices. When considering the useful life period, the failure rate λ is constant and is related to the reliability $R(t)$ by the following equation [1]:

$$R(t) = e^{-\lambda t} \tag{4.1}$$

In addition, for a constant failure rate λ the Mean-Time-To-Failure (MTTF) is given by the following equation [1]:

$$MTTF = \frac{1}{\lambda} \tag{4.2}$$

Here, the reliability function $R(t)$ corresponds to an exponential failure time distribution. For high scale reliability tests of NAND Flash memories in the useful life period, if r memories over N are defective after a test time t_{test}, then the number of failures in the time interval t_{test} follows a Poisson distribution with a parameter λt_{test}. The Poisson process for failure occurrence event is valid when r value is relatively small compared to N. The probability of r memories failure occurrence is then given by [78]:

$$P(\lambda t_{test}, r) = \frac{(\lambda t_{test})^r e^{-(\lambda t_{test})}}{r!} \tag{4.3}$$

Let CL be a chosen confidence level. Starting from (4.3), an upper value of λ guaranteeing this confidence level λ_{up} can be deduced from the following equations [78]:

$$\begin{aligned} CL &= 1 - \sum_{i=0}^{r} \frac{(\lambda t_{test})^i e^{-(\lambda t_{test})}}{i!} \\ &= 1 - \sum_{i=0}^{r} \frac{x^i e^{-x}}{i!} \end{aligned} \tag{4.4}$$

where $x = \lambda t_{test}$. λ_{up} can be deduced from the random variable X and its upper bound related to CL by the cumulative distribution function P(X<x). In fact, considering a Gamma distribution $Y \backsim Gamma(\alpha, \beta)$, its cumulative distribution function is given by P(Y<y):

$$P(Y < y) = F(y, \alpha, \beta) = 1 - \sum_{i=0}^{\alpha-1} \frac{(\beta y)^i e^{-(\beta y)}}{i!} \tag{4.5}$$

Combining (4.4) and (4.5), one can see that x follows the Gamma distribution $X \backsim Gamma(r+1, 1)$. Thus, using the properties of a Gamma random variable and the chi-squared distribution, a special case of the Gamma distribution, we have [78]:

$$\begin{aligned} 2X &\backsim Gamma(r+1, 2) \\ &\backsim \chi^2_{2(r+1)} \end{aligned} \tag{4.6}$$

Considering, the confidence level CL and the relation $x = \lambda t_{test}$, the upper bound of λ is defined by (4.7) [71][78]. The values of χ^2 are derived from probability tables or by the application of approximation equations [79]:

$$\lambda_{up} = \frac{1}{MTTF} = \frac{\chi^2_{CL,2(r+1)}}{2t_{test}} \tag{4.7}$$

For a Gamma distribution, the cumulative distribution function (CDF) is also expressed as the integral of the probability density function (PDF). Having $x = \lambda t_{test}$ and the Gamma distribution PDF of Fig. 4.3, CL corresponds to the percent area under the PDF curve limited by $\frac{\chi^2_{CL,2(r+1)}}{2} = \lambda_{up} t_{test}$ [71][79].

In the above, high scale reliability tests have been considered. Here we study the case of one MLC NAND Flash memory. As each memory array is composed of several blocks and each block of

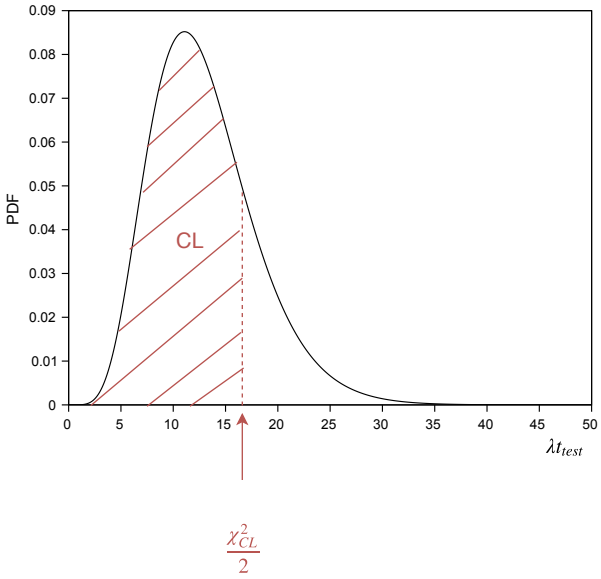

Figure 4.3: PDF of Gamma distribution and estimation of λ_{up} value corresponding to a confidence level CL

numerous pages, a similar statistical approach can be applied to each memory page of the memory.

Here, only retention errors are considered as they are dominant over disturb errors and characterized by a rate that increases with the retention age. For a memory page with N_{vul} bits vulnerable to retention errors and N_{ret} retention errors occurring by the retention age t_{age}, the above relations based on exponential chi-squared and Poisson distributions are no longer valid. N_{ret} is comparable to N_{vul} and the probability of retention errors occurrence by the retention age t_{age} can no longer be approximated by a Poisson law. A binomial law is used instead to estimate the probability of retention errors occurrence in the memory page with N_{vul} vulnerable bits and after a storage period t_{age} (4.8).

$$P(N_{vul}, N_{ret}, t_{age}, \lambda) = \binom{N_{vul}}{N_{ret}} e^{-\lambda t_{age}(N_{vul}-N_{ret})}(1 - e^{-\lambda t_{age}})^{N_{ret}} \tag{4.8}$$

With the use of the binomial law, (4.8) is seen as a discrete probability distribution of retention errors N_{ret}. Using an appropriate normalization factor, a continuous probability distribution of λ can be deduced from (4.8) as shown in (4.9) and used to determine a λ upper value of λ_{up} for a given confidence level CL (4.10).

$$\int_0^{\infty} P(N_{vul}, N_{ret}, t_{age}, \lambda) d\lambda = \frac{1}{t_{age}(N_{vul} - N_{ret})} \tag{4.9}$$

$$CL = (N_{vul} - N_{ret})\binom{N_{vul}}{N_{ret}} \int_{X_{up}}^{1} X^{(N_{vul}-N_{ret}-1)}(1-X)^{N_{ret}} dX \tag{4.10}$$

with $X = e^{-\lambda t_{age}}$ and $X_{up} = e^{-\lambda_{up} t_{age}}$.

Since reliability is defined by (4.1), the unreliability function of the considered MLC NAND Flash corresponds to the retention raw bit error rate. Retention RBER ($RBER_{ret}$), which is the probability of a vulnerable bit to be affected by a retention error, is given by the following equation:

$$RBER_{ret} = 1 - e^{-\lambda t_{age}} \tag{4.11}$$

where λ varies from one page to another and its upper value for a given CL is deduced from (4.10). This definition of $RBER_{ret}$ is in agreement with results from [14] that give RBER evolution with retention time for cycled MLC NAND Flash memories from different manufacturers (Fig. 4.4).

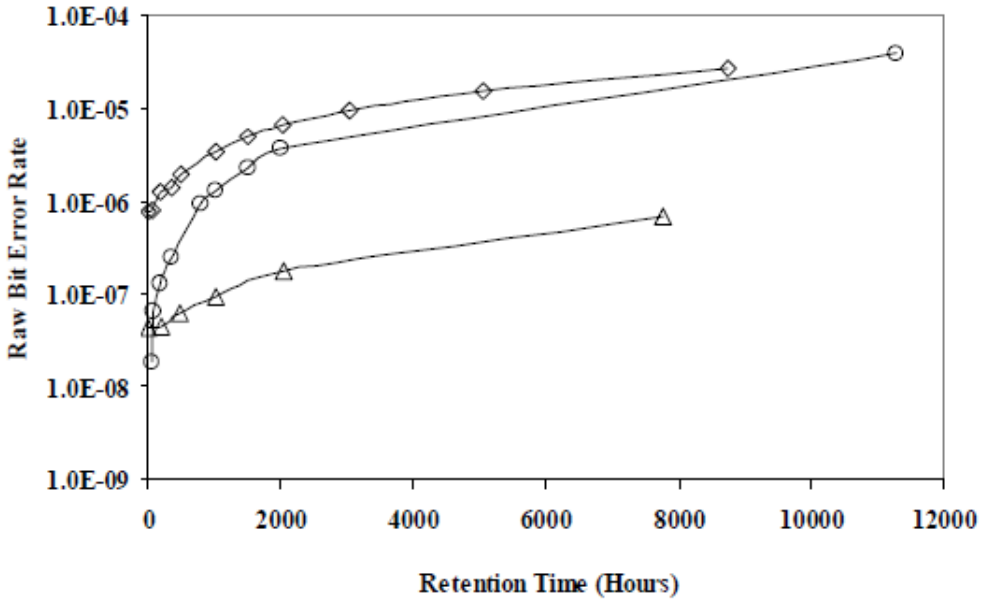

Figure 4.4: RBER variation with retention time at room temperature for 10K cycled MLC Flash from different vendors [14]

4.2 Statistical approach for tolerated RBER improvement

4.2.1 Left retention time estimation

In this section, a statistical approach is proposed to improve the tolerated RBER in MLC NAND Flash memories of enterprise-class SSDs for a specific ECC strength and a fixed maximum value

of UBER. Thus, reliability is improved without the use of a strong ECC or the consideration of worst-case refreshes. This approach is based on the application of the statistical method explained in section 4.1.2 to the different memory pages. Check operations of each memory page are used to estimate the λ parameter in (4.11) and τ the reliable left retention time. τ is the maximum remaining storage time in each page that guarantees to not exceed the target UBER value. Here, MLC NAND Flash memories of enterprise-class SSD are considered. The target UBER value is chosen to be equal to 10^{-16} [68]. We also chose a maximum data storage period of 36 months. 3 years is the target retention time as this value is reasonable for the considered enterprise-class SSDs in this study.

At each check of a memory page, the left retention time is estimated based on the number of detected retention errors N_{ret} and the retention age t_{age}. t_{check} is the maximum time period between two consecutive check operations. If the left retention time that guarantees the integrity of information until the next check operation is smaller than t_{check}, then data needs to be refreshed. The steps of algorithm 1 need to be performed at each check of a Flash memory page.

Algorithm 1: Check operations based on embedded statistics

Known parameter: M, strength of ECC protecting a Flash memory page

Known parameter: N_{vul}, initial number of bits vulnerable to retention errors in the page

Known parameter: N_{ret}, number of retention errors in the page when check is performed

Known parameter: N_{non_ret}, number of non-retention errors in the page at the check time

Known parameter: t_{check}, maximum period of time between consecutive checks

Calculate the retention age t_{age} of the page since last program or refresh operation

Estimate the left retention time τ that guarantees the target UBER as a function of t_{age}, N_{ret}, N_{non_ret} and N_{vul}

if $\tau < t_{check}$ **then** refresh the accessed page

end

These operations require the prior knowledge of the different parameters M, N_{ret}, N_{non_ret}, N_{vul} and t_{age}. M corresponds to the strength of the used ECC. In our study, M designates the maximum number of errors that can be corrected in a read page with the available ECC. N_{ret} and N_{non_ret} can be calculated by using the ECC decoder. In fact, when the decoding step is executed, a search for potential errors in each bit location is performed. When an erroneous bit location is found, the error polarity allows to identify the error type by using the fingerprints of Table 4.1. Thus, the number of retention and non-retention errors in the read page can be counted when a check operation is performed. Here, we consider a small number of non-retention errors compared

to the number of predominant retention errors.

The number of vulnerable bits N_{vul} corresponds to the initial number of bits in a memory page that can potentially be affected by a retention error. It can be saved as a metadata for each considered memory page upon programming. It corresponds to the number of memory cells, of the same logic memory page, with initial programmed states (P1, P2 or P3 Fig. 4.1). These cells can potentially loose their charge and are vulnerable to retention errors. An easier way to deal with the check operations at each read is to consider that all the bits on a memory page are vulnerable bits N_{vul}. This consists in a pessimistic but easy way to handle the value of N_{vul}. The overhead of estimating and storing the value of N_{vul} for each memory page is avoided. This way, the left retention time τ estimation in each memory page is only calculated in reference to the maximum value of N_{vul}.

The retention age t_{age} can be estimated with the use of timestamps. t_{age} can then be calculated as the difference between: (a) a current state of a timer that provides timestamps when the page is accessed for a check operation and (b) the initial value of the timestamp when the page was programmed or refreshed for the last time. A single timer can be used for all pages of the same memory block. In this case, t_{age} will be the same for all the block pages and it corresponds to the t_{age} of the first page to be programmed in the block. This allows to highly reduce the overhead of timestamps storage.

The reliable left retention time τ can be estimated off-line for the different combinations of parameters t_{age}, N_{vul}, N_{ret} and N_{non_ret}. τ calculation is based on the estimation of the λ parameter in (4.11) by the statistical approach of section 4.1.2. Considering algorithm 1, when a check operation is performed and parameters t_{age}, N_{ret}, N_{non_ret} and N_{vul} are known, retention RBER $RBER_{ret}$ is calculated by using the statistically estimated λ value. Assuming that retention errors are the only type of errors that accumulate in time and that retention and non-retention errors are handled by the ECC at each check operation, $RBER_{ret}$ is then used to estimate UBER as detailed in Appendix A and Appendix B. The maximum future reliable storage period is calculated in reference to the upper value of UBER, i.e. 10^{-16}, and is equal to τ.

In our study, we first estimated the reliable left retention time τ as a function of the retention errors number at different retention ages ranging from 6 to 36 months. The considered pages have a size of 16 Kb. For the graph of Fig. 4.5, a number of vulnerable bits N_{vul} equal to 16 Kb is chosen. Here, the left retention time τ is calculated with a granularity of 1 month. The chosen confidence level CL for the statistical estimation of λ is equal to 90%.

Fig. 4.5 shows that the reliable left retention time τ rapidly decreases with the number of retention errors at a given retention age. In fact, with a higher number of retention errors at a given t_{age}, UBER is more likely to quickly reach the limit value of 10^{-16}. τ increases with the retention age for a given number of retention errors. Later the retention errors appear, more time UBER takes to reach the limit value of 10^{-16}.

As explained above, the overhead of the estimation and storing of N_{vul} value for each memory page can be avoided by considering its maximum value. This value can be chosen equal to the

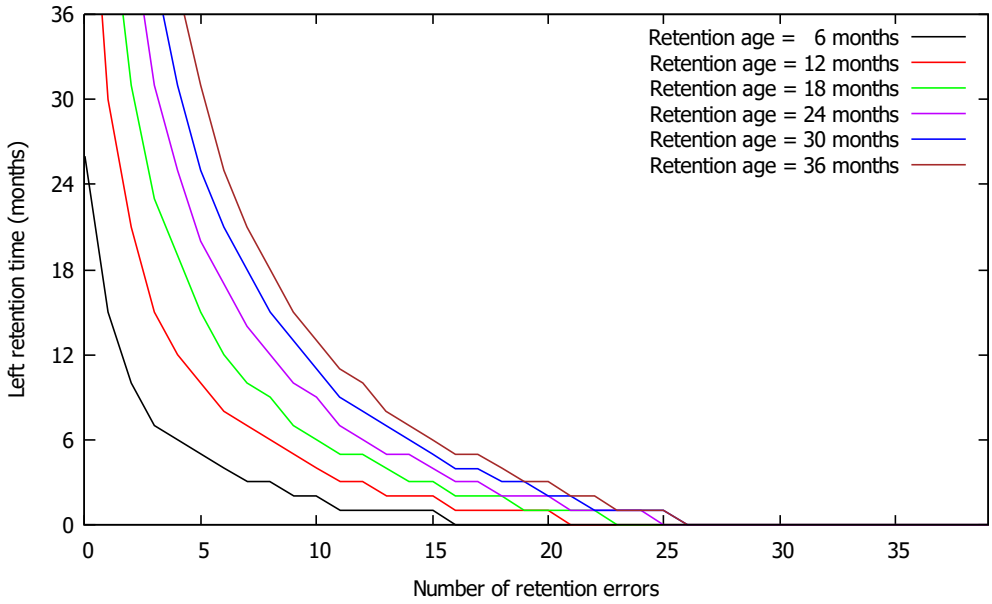

Figure 4.5: Reliable left retention time as a function of N_{ret} at different retention ages

considered page size. From the expression of UBER in Appendix A, UBER monotonically increases with $RBER_{ret}$ and also with N_{vul} for higher values of $RBER_{ret}$. The upper limit of UBER is more rapidly reached for higher values of N_{vul}, so τ decreases with N_{vul}. This is illustrated in Fig. 4.6 that shows the evolution of τ as a function of the number of retention errors N_{ret} for different values of N_{vul}. Here, the considered retention age is equal to 36 months, the target retention time. In this case, we have the largest values of $RBER_{ret}$, the largest remaining retention times τ (Fig. 4.5) and the largest differences between the values of τ for different N_{vul}.

Another important metric in algorithm 1 is t_{check}, the maximum period of time separating two consecutive check operations of the memory pages. t_{check} fixes the granularity with which the memory pages are refreshed in case the reliable left retention time is not enough to reach the next check operation. In our study, different t_{check} values are used to estimate the improvement in the tolerable retention RBER that guarantees the upper limit of UBER. In the following, it is shown that for t_{check} of few months, the tolerable retention RBER is highly improved. This order of magnitude of t_{check} is relatively large, which results in a very small impact on the average response time of the considered SSD. Besides, not all the check operations of algorithm 1 result in a refresh. In addition, redundant verification steps can be avoided by taking advantage of functional reads in Flash memories to run the operations of algorithm 1. In this case, one bit of metadata can be reserved to keep a trace of the last refresh due to a functional read.

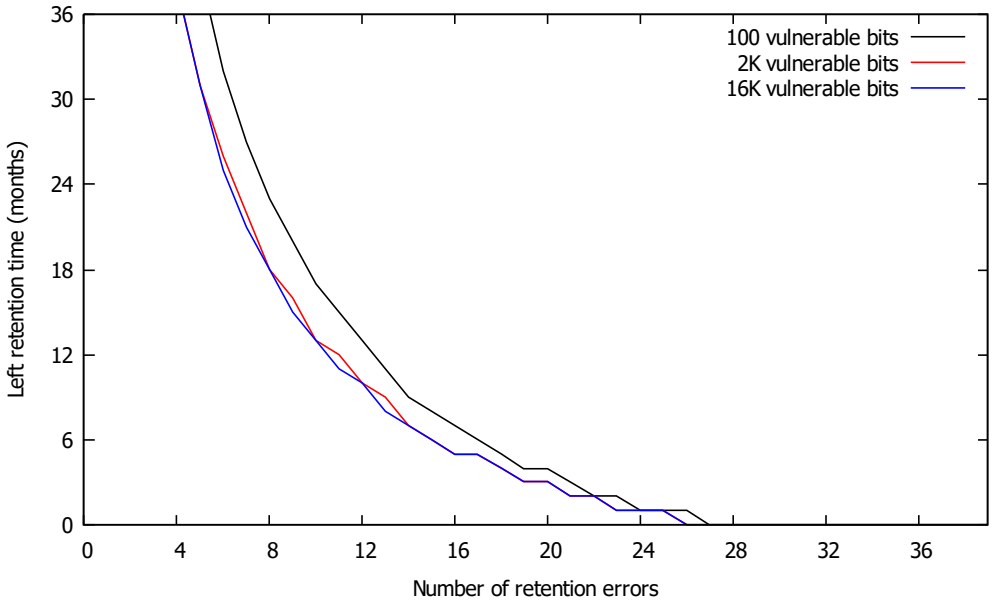

Figure 4.6: Reliable left retention time as a function of N_{ret} at 36 months of retention age for different values of N_{vul}

4.2.2 Maximum tolerated retention RBER estimation

Reliability improvement is evaluated in NAND Flash pages where the check operations of algorithm 1 are executed. While respecting the upper limit of UBER at target retention time, the increase in the maximum tolerated retention RBER is evaluated and compared to a no-check scheme that respects the same condition for UBER. For this evaluation, we consider 16 Kb NAND Flash memory pages with ECCs able to correct up to 10, 20, 30 and 40 single-bit errors per page. All the bits of the page are considered vulnerable to retention errors (N_{vul} = 16 Kb) and 1 non-retention error affects the considered page. The used method to calculate UBER is given in appendix A and B. The maximum tolerated retention RBER is estimated in relation to the left retention time τ of algorithm 1 with a considered confidence level of 90%. Results are summarized in Table 4.2 and Table 4.3. Different t_{check} periods have been considered, ranging from 1 month to 6 months.

Table 4.2 shows that the maximum tolerated $RBER_{ret}$ increases when the check period t_{check} is reduced. This can be explained by the fact that more frequent check and refresh operations allow a more precise estimation of τ. Refresh operations are better adapted to the state of data in NAND Flash pages. As a consequence, tolerated $RBER_{ret}$ can be increased without exceeding the upper limit of UBER. Table 4.3 shows that for a check period t_{check} of 1 month, tolerable $RBER_{ret}$ is improved by a factor that varies between 19x and 28x. In addition, a 1 month check period allows a better improvement of the tolerable $RBER_{ret}$ than the use of a three times stronger ECC. In fact, the tolerable $RBER_{ret}$ for t_{check} = 1 month and 10 correctable bits per page is slightly higher than

Number of correctable errors per page	Maximum tolerated $RBER_{ret}$ with $UBER \leqslant 10^{-16}$					
	No check	Check period t_{check}				
		1 month	2 months	3 months	4 months	6 months
40	$6.28x10^{-4}$	$1.53x10^{-2}$	$7.70x10^{-3}$	$5.14x10^{-3}$	$3.86x10^{-3}$	$2.61x10^{-3}$
30	$3.60x10^{-4}$	$9.69x10^{-3}$	$4.86x10^{-3}$	$3.24x10^{-3}$	$2.44x10^{-3}$	$1.71x10^{-3}$
20	$1.46x10^{-4}$	$4.10x10^{-3}$	$2.05x10^{-3}$	$1.40x10^{-3}$	$1.09x10^{-3}$	$7.83x10^{-4}$
10	$1.89x10^{-5}$	$3.73x10^{-4}$	$1.91x10^{-4}$	$1.32x10^{-4}$	$1.02x10^{-4}$	$7.23x10^{-5}$

Table 4.2: Maximum tolerated $RBER_{ret}$ in NAND Flash pages when $UBER \leqslant 10^{-16}$

Number of correctable errors per page	$RBER_{ret}$ improvement factor compared to a no check case				
	Check period t_{check}				
	1 month	2 months	3 months	4 months	6 months
40	24.4	12.3	8.2	6.1	4.2
30	26.9	13.5	9.0	6.8	4.8
20	28.1	14.0	9.6	7.5	5.4
10	19.7	10.1	7.0	5.4	3.8

Table 4.3: Maximum tolerated $RBER_{ret}$ improvement ratio compared to a no check case in NAND Flash pages

the one with no check operations and 30 correctable bits per page. In the same way, a 6 months check period allows a better improvement of the tolerable $RBER_{ret}$ than the use of a two times stronger ECC. In fact, the tolerable $RBER_{ret}$ for $t_{check} = 6$ months and 20 correctable bits per page is slightly higher than the one with no check operations and 40 correctable bits per page. This proves that the used statistical approach for the estimation of reliable left retention time is more effective than the use of a stronger ECC. This prevents the storage and latency overhead introduced by the use of a stronger ECC. It should be noted that better results relative to the improvement of the maximum tolerated $RBER_{ret}$ can be obtained with smaller check periods. In addition, in Table 4.3, improvement factors are rather pessimistic as the functional reads that can be used for check operations of algorithm 1 are not taken into account.

The results reported in Table 4.2 and Table 4.3 are for SSDs with power-off periods much smaller than the check period t_{check} which is the case of enterprise-class SSDs [74]. This guarantees the non-alteration of data at power-off periods. In the other cases, the power-off period t_{power_off} needs to be taken into account in the check operations of algorithm 1. The left retention time τ needs to be compared to $t_{check}+t_{power\_off}$ and refreshes need to be triggered in case $\tau < t_{check}+t_{power\_off}$.

It should be noted that the data refresh frequency resulting from the use of algorithm 1 is not correlated to the check frequency. A refresh is only triggered when the *if condition* in algorithm 1 is verified. The refresh frequency is only related to the $RBER_{ret}$ calculated via the estimation of τ. The improvements in the maximum tolerated $RBER_{ret}$ are based on the refresh operations triggered by algorithm 1. The number of these refresh operations is highly reduced compared to the number of refresh operations of a conventional scheme with systematic periodic refreshes that guarantees the same protection level. Fig 4.7 shows the probability of refresh operations triggered by algorithm 1 for a target storage period of 36 months.

Fig. 4.7 shows that refresh probability is very small when $RBER_{ret}$ can still be manageable by the ECC alone without the check operations of algorithm 1. Refresh probability increases starting from the maximum $RBER_{ret}$ value that can be managed by the ECC only and continues to increase for larger values of $RBER_{ret}$. With the smallest check period, we have the lowest values of refresh probability as it is easier to guarantee data integrity until the following check operation. This proves that the refresh probability is effectively correlated to the evolution of the actual $RBER_{ret}$.

The average period of time between refresh operation triggered by algorithm 1 is illustrated in Fig. 4.8. The ideal refresh period of the systematic refresh scheme is also illustrated for comparison. For the lowest check period (1 month), we have the highest values of the average refresh period and the biggest reduction in the number of refresh operations compared to a systematic refresh scheme.

4.3 Selective refresh with retention RBER linear approximation

Here, the statistical approach explained in section 4.1.2 is simplified by a linear approximation of the retention RBER in (4.11). The same approach of check operations in algorithm 1 is used to estimate the reliable left retention time τ before the target UBER value is exceeded. Similarly, this

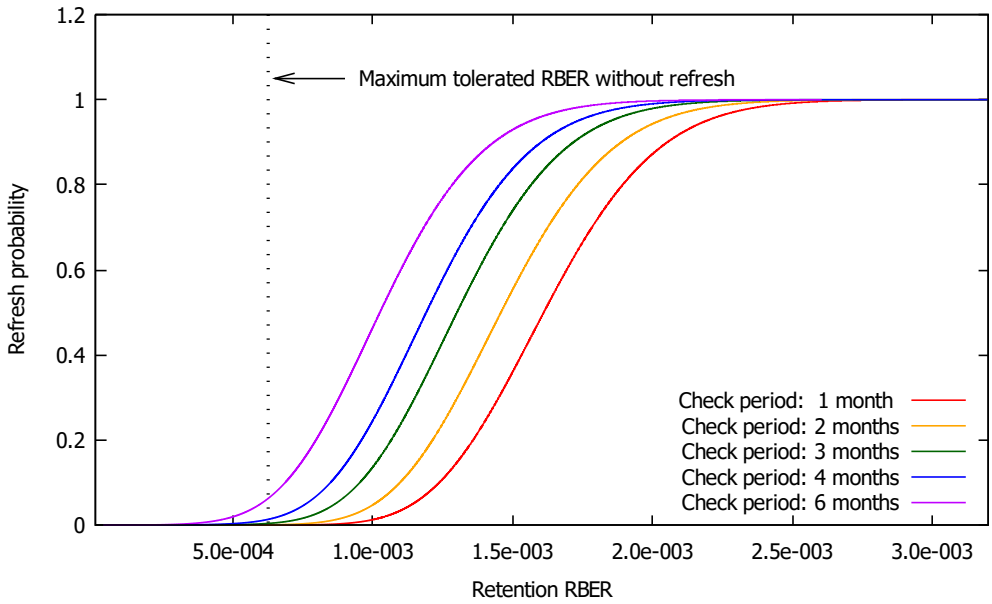

Figure 4.7: Refresh probability over a target storage period of 36 months for the refresh scheme of algorithm 1. Flash memory pages with 16 Kb vulnerable bits and an ECC able to correct up to 40 bits

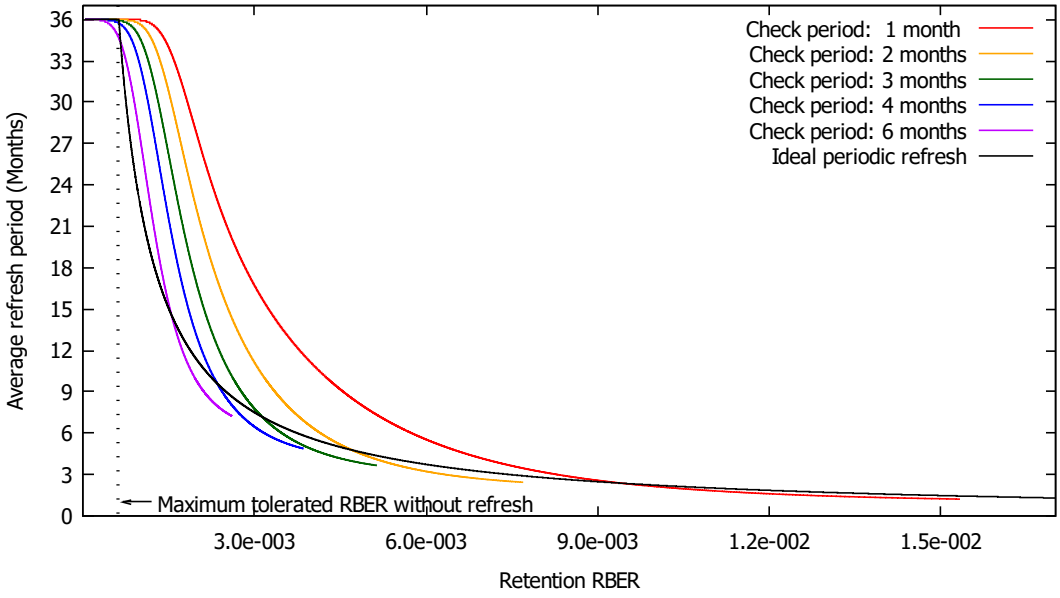

Figure 4.8: Average time between refresh operations for the refresh scheme of algorithm 1 in comparison to a systematic refresh scheme with a fixed refresh frequency. Flash memory pages with 16 Kb vulnerable bits and an ECC able to correct up to 40 bits

approach allows to manage the RBER variations beyond the limits imposed by the provided ECC. A selective refresh is performed, when the left retention time is smaller that a fixed check period. Functional reads can also be used to perform the check operations, which contributes to adapt the refresh frequency to the actual retention RBER, in addition to limiting the performance overhead related to this refresh scheme.

4.3.1 Left retention time estimation with retention RBER linear approximation

Starting from the expression of $RBER_{ret}$ in (4.11), a linear approximation can be done in order to estimate the left retention time τ in a simpler way than in the case of the statistical approach. This approximation is based on noting that λt_{age} is negligible in comparison to 1 due to the orders of magnitude of λ. In this case, retention RBER is expressed as follows:

$$RBER_{ret} = \lambda t_{age} \tag{4.12}$$

Retention RBER is the probability that a vulnerable bit is affected by a retention error. The relation in (4.13) can then be deduced from the approximation of (4.12) when considering M the maximum number of errors that the ECC can correct per read page, N_{ret} the number of retention errors at retention age t_{age} and N_{non_ret} the number of already existing non-retention errors in the considered page.

$$\frac{N_{ret}}{M - N_{non\_ret}} = \frac{t_{age}}{t_{age} + \tau} \tag{4.13}$$

The reliable left retention time τ can then be estimated as follows:

$$\tau = \alpha t_{age}\left(\frac{M - N_{non\_ret}}{N_{ret}} - 1\right) \tag{4.14}$$

where α is a factor that allows to take into account the statistical variations of N_{ret} in the expression of τ.

Compared to the statistical approach of the previous section, the expression of τ with the linear approximation of $RBER_{ret}$ does not show any dependence on λ or on N_{vul}. Thus, the statistical estimation of λ is avoided and τ only depends on N_{ret}, N_{non_ret} and t_{age}. Here, we consider that if the number of retention errors at t_{age} is equal to zero, then τ is chosen to be equal to the target retention time which is 3 years. Similarly to the statistical approach, N_{ret} and N_{non_ret} are calculated with the help of the ECC decoder. t_{age} is calculated with a timer able to provide timestamps.

Considering a maximum time interval t_{check} between check operations, refresh is triggered through the calculation of τ and by the application of the steps of algorithm 2. This algorithm is very similar to algorithm 1 used for the refreshes of the statistical scheme. The only difference consists in the calculation of τ.

Algorithm 2: Check operations based on the linear approximation of $RBER_{ret}$

Known parameter: M, strength of ECC protecting a Flash memory page

Known parameter: N_{ret}, number of retention errors in the page when check is performed

Known parameter: N_{non_ret}, number of non-retention errors in the page at the check time

Known parameter: t_{check}, maximum period of time between consecutive checks

Calculate the retention age t_{age} of the page since last program or refresh operation

Estimate the remaining retention time τ that guarantees the target UBER as a function of t_{age}, N_{ret} and N_{non_ret}

if $\tau < t_{check}$ **then** refresh the accessed page

end

The remaining retention time can be either computed on or off line for the different combinations of N_{ret}, N_{non_ret} and t_{age}. (4.14) shows that τ decreases with N_{ret}. Another implementation of algorithm 2 that allows the storage overhead reduction would consist in changing the *if condition* of algorithm 2 by a comparison between the measured number of N_{ret} and a maximum value of N_{ret}. This maximum value would correspond to the largest value of N_{ret} that gives the lowest left retention time guaranteeing the integrity of data till the following check. This maximum value of tolerable N_{ret} would increase with the increase of the ECC strength and when t_{check} deceases.

4.3.2 Selective refresh optimization

The check operations described by both algorithms 1 and 2 are done at a fixed frequency imposed by the means of warnings. These warnings can be triggered by a conventional timer or a timer that takes into account temperature variations, as the A-timer of Chapter 3. In addition, check operations can be done at the functional reads of memory pages. Thus, frequently accessed pages may be skipped in the periodic checks triggered by the chosen timer.

This can be done by the means of a single bit flag per accessed page that is used to indicate whether the page has already been checked or not. The flag is initialized to 1 and it is set to 0 each time the page is read, programmed or refreshed. When the first subsequent warning of the timer comes, the page is not checked, only its flag is updated to 1. On the following warning, the page has to be checked and its flag is updated to 0, unless a functional read, program or refresh occurred since the last warning. In this case, no check is done and only the flag is set to 1. In order to insure the success of this method based on a timer use, the warning period t_{warn} should be set to half the check period ($t_{check} = 2t_{warn}$). Fig. 4.9 shows how page flags are managed in order to reduce the number of check operations of algorithms 1 and 2.

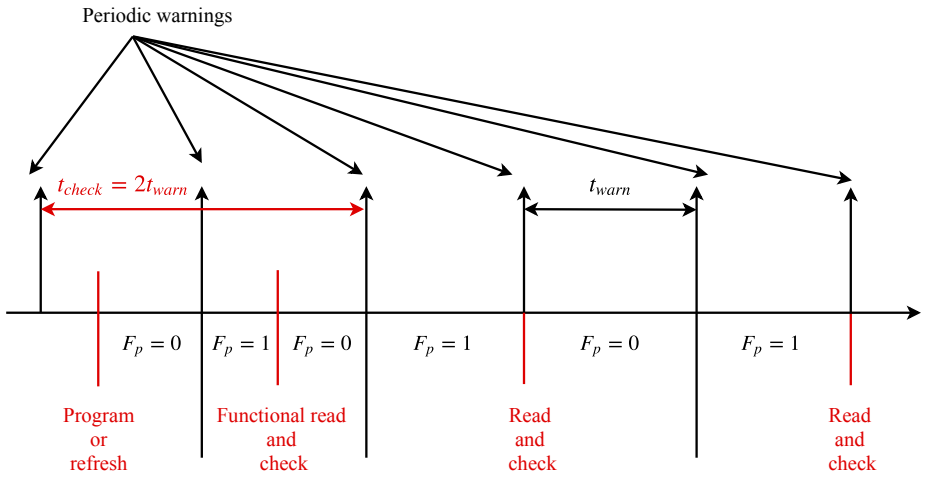

Figure 4.9: Management of page flags in order to reduce the number of check operations of algorithms 1 and 2

4.3.3 Improvement in the maximum tolerated retention RBER

Similarly to the statistical approach, reliability improvement is evaluated in NAND Flash pages where the check operations of algorithm 2 are executed. The increase in the maximum tolerated retention RBER, while respecting the upper limit of UBER at target retention time, is estimated. The retention RBER improvement is evaluated in comparison to a no check scheme that respects the same condition for UBER. For this evaluation, we consider 16 Kb NAND Flash memory pages with ECCs able to correct up to 10, 20, 30 and 40 single-bit errors per page. Here, no non-retention errors are considered and all the bits of the page are vulnerable to retention errors (N_{vul} = 16 Kb). The same method as for the statistical scheme is used to calculate UBER and is given in Appendix A and Appendix B. Results of the maximum tolerated retention RBER in relation to the left retention time τ of algorithm 2 are summarized in Table 4.4 and Table 4.5. Different t_{check} periods have been considered, ranging from 1 month to 6 months.

As for the statistical scheme, Table 4.4 shows that the maximum tolerated $RBER_{ret}$ increases when the check period t_{check} is reduced. Table 4.5 shows that for a check period t_{check} of 1 month, tolerable $RBER_{ret}$ is improved by a factor that varies between 32x and 35x. In addition, a 1 month check period allows a better improvement of the tolerable $RBER_{ret}$ than the use of a four times stronger ECC. In fact, the tolerable $RBER_{ret}$ for t_{check} = 1 month and 10 correctable bits per page is slightly higher than the one with no check operations and 40 correctable bits per page. In the same way, a 6 months check period allows a better improvement of the tolerable $RBER_{ret}$ than the use of a two times stronger ECC. In fact, the tolerable $RBER_{ret}$ for t_{check} = 6 months and 20

Number of correctable errors per page	Maximum tolerated $RBER_{ret}$ with $UBER \leqslant 10^{-16}$					
	No check	Check period t_{check}				
		1 month	2 months	3 months	4 months	6 months
40	$6.56x10^{-4}$	$2.31x10^{-2}$	$1.16x10^{-2}$	$7.76x10^{-3}$	$5.82x10^{-3}$	$3.89x10^{-3}$
30	$3.84x10^{-4}$	$1.32x10^{-2}$	$6.63x10^{-3}$	$4.42x10^{-3}$	$3.32x10^{-3}$	$2.21x10^{-3}$
20	$1.56x10^{-4}$	$5.65x10^{-3}$	$2.83x10^{-3}$	$1.89x10^{-3}$	$1.42x10^{-3}$	$9.62x10^{-4}$
10	$2.64x10^{-5}$	$8.52x10^{-4}$	$4.26x10^{-4}$	$2.85x10^{-4}$	$2.14x10^{-4}$	$1.44x10^{-4}$

Table 4.4: Maximum tolerated $RBER_{ret}$ in NAND Flash pages when $UBER \leqslant 10^{-16}$

Number of correctable errors per page	$RBER_{ret}$ improvement factor compared to a no check case				
	Check period t_{check}				
	1 month	2 months	3 months	4 months	6 months
40	35.2	17.7	11.8	8.9	5.9
30	34.4	17.3	11.5	8.6	5.8
20	34.2	17.2	11.5	8.6	5.8
10	32.3	16.1	10.8	8.1	5.5

Table 4.5: Maximum tolerated $RBER_{ret}$ improvement ratio compared to a no check case in NAND Flash pages

correctable bits per page is slightly higher than the one with no check operations and 40 correctable bits per page. This proves that the used linear approximation of retention RBER approach for the estimation of the reliable remaining retention time is more effective than the use of a stronger ECC. In Table 4.5, improvement factors are rather pessimistic as the functional reads that can be used for check operations of algorithm 2 are not taken into account.

With the improvement of retention RBER comes a reduction of the number of refresh operations in comparison to a conventional scheme with systematic refresh operations. Thus, the average time between refresh operations is increased when compared to the ideal refresh period of a systematic refresh scheme as illustrated in Fig. 4.10.

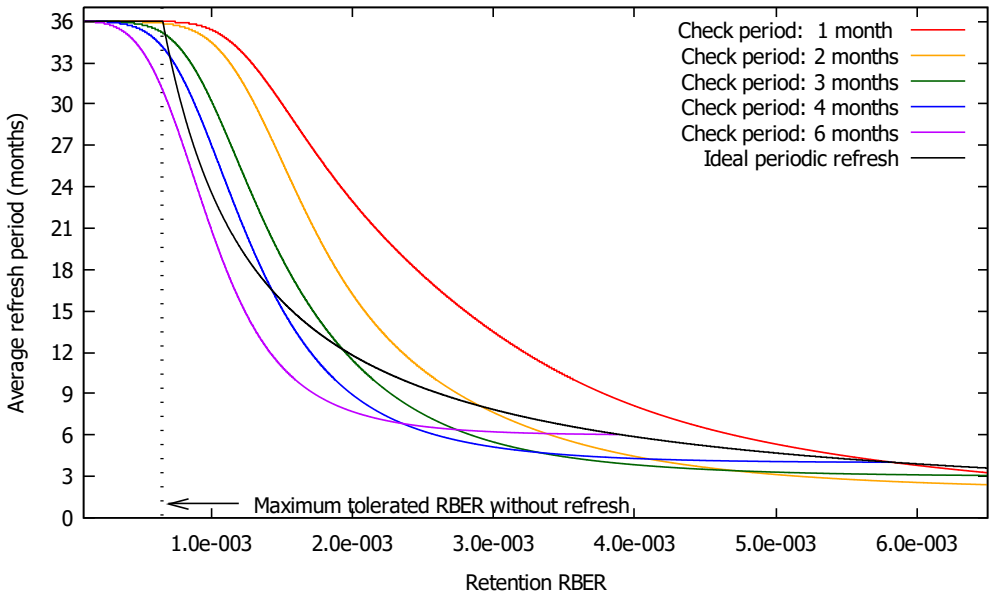

Figure 4.10: Average time between refresh operations for the refresh scheme of algorithm 2 in comparison to a systematic refresh scheme with a fixed refresh frequency. Flash memory pages with 16 Kb vulnerable bits and an ECC able to correct up to 40 bits

As the average period of time between refreshes is increased, the Flash access time for read and write relative to refresh operations is reduced. The reduction of the access time can be calculated as follows:

$$\frac{(\tau_{wr}+\tau_{rd})D_{sys\_refresh}f_{sys\_refresh}}{\tau_{wr}D_{refresh}f_{lin\_refresh}+\tau_{rd}D_{check}f_{lin\_check}} \tag{4.15}$$

where τ_{wr} and τ_{rd} are respectively the page write and read latencies. $f_{sys\_refresh}$ is the fixed refresh frequency in the case a systematic refresh scheme. $D_{sys\_refresh}$ is the amount of data that needs to be systematically refreshed. f_{lin_check} and D_{check} are respectively the frequency of

check operations and the amount of data checked with the scheme of algorithm 2. $f_{lin\_refresh}$ and $D_{refresh}$ are respectively the frequency of refresh operations and the amount of data refreshed with the scheme of algorithm 2. In our simulations, the α factor in (4.14) is chosen such that the upper limit of UBER is reached for a retention RBER that guarantees an average retention time equal to $t_{check} = \frac{1}{f_{lin\_check}}$. In this case f_{check} and $f_{sys\_refresh}$ are equal, D_{check} and $D_{sys\_refresh}$ are also equal. As the amounts of data actually refreshed is smaller that the checked data, a lower bound for (4.15) can be expressed as follows:

$$\frac{(\tau_{wr} + \tau_{rd}) f_{sys\_refresh}}{\tau_{wr} f_{lin\_refresh} + \tau_{rd} f_{lin\_check}} \tag{4.16}$$

Fig. 4.11, 4.12, 4.13 and 4.14 illustrate the reduction of access time for an MLC NAND Flash with read and write latencies equal to 975 μs and 50 μs. This graphs show the access time reductions for different ECC strengths with respect to a fixed-frequency systematic refresh scheme. The systematic refresh scheme allows to tolerate the same values of maximum retention RBER guaranteed by the proposed scheme of algorithm 2. The highest values of access time reductions are observed for retention RBER values that can be managed by the available ECC only.

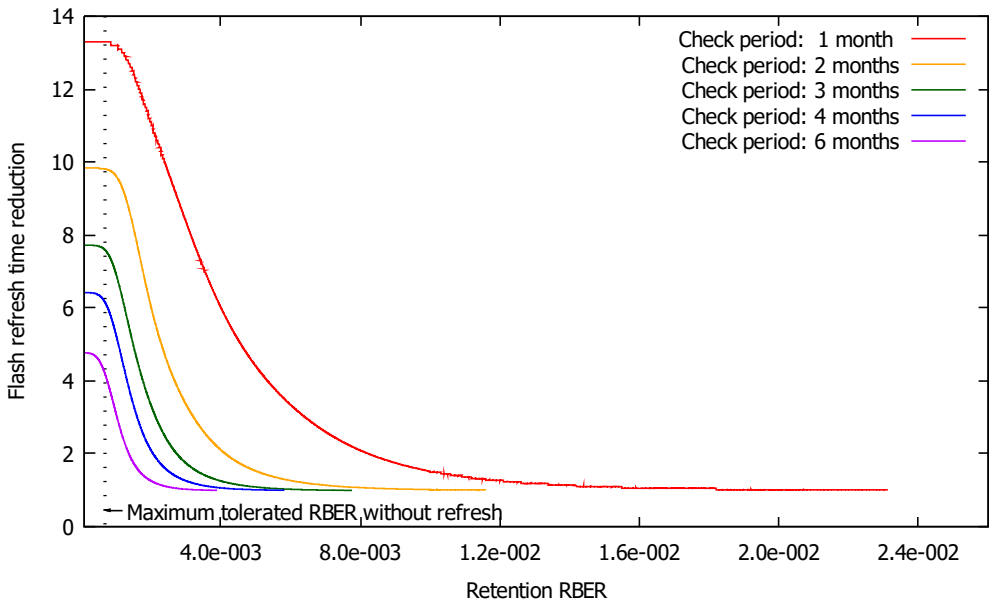

Figure 4.11: Access time reductions with respect to a systematic refresh scheme with a fixed refresh frequency. Flash memory pages with 16 Kb vulnerable bits and an ECC able to correct up to 40 bits

Fig. 4.14 shows a different shape of the access time reduction curve compared to the case of other ECC strengths. This difference can be explained by the fact that at 10 correctable errors per

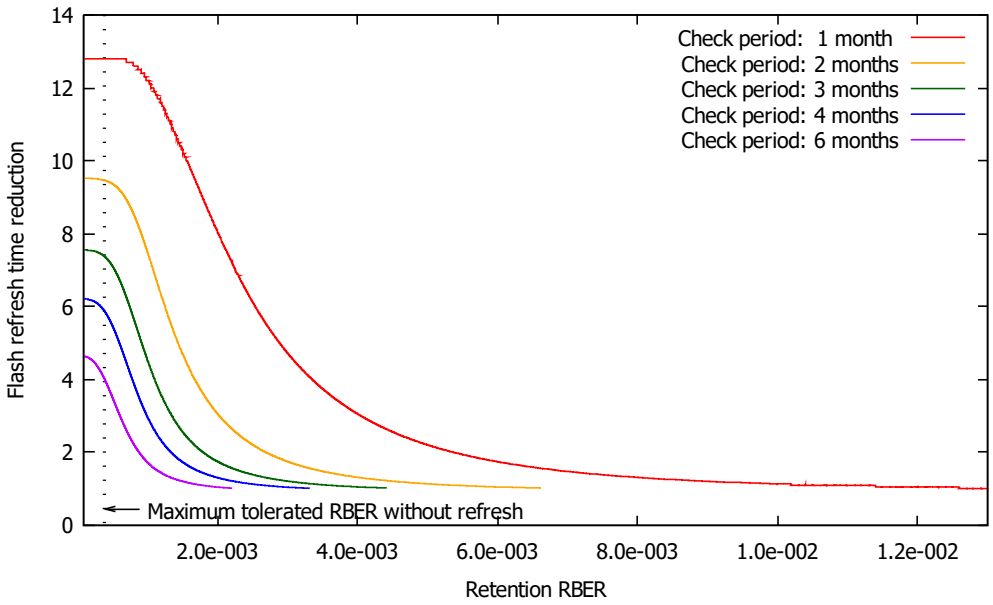

Figure 4.12: Access time reductions with respect to a systematic refresh scheme with a fixed refresh frequency. Flash memory pages with 16 Kb vulnerable bits and an ECC able to correct up to 30 bits

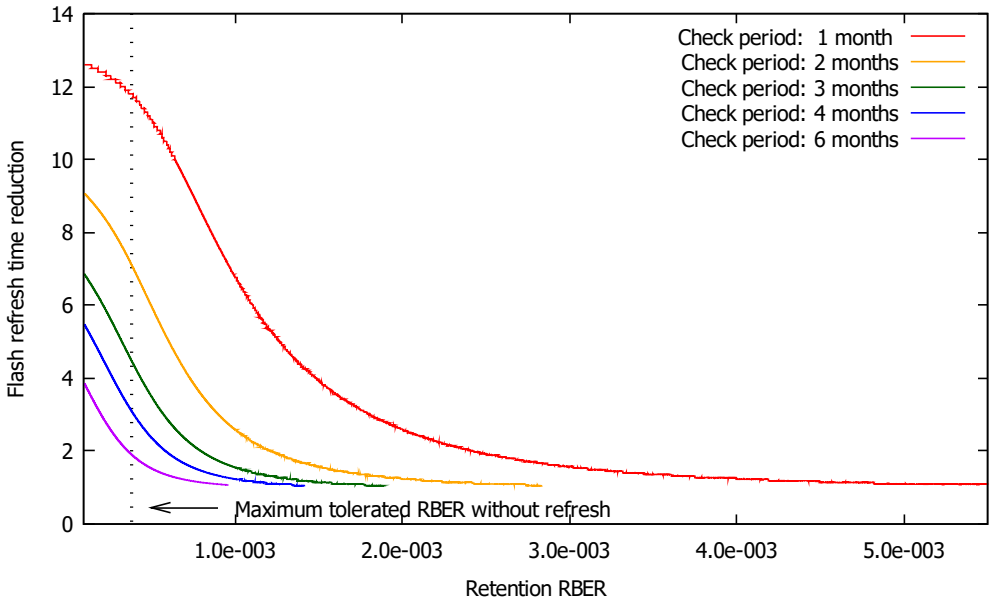

Figure 4.13: Access time reductions with respect to a systematic refresh scheme with a fixed refresh frequency. Flash memory pages with 16 Kb vulnerable bits and an ECC able to correct up to 20 bits

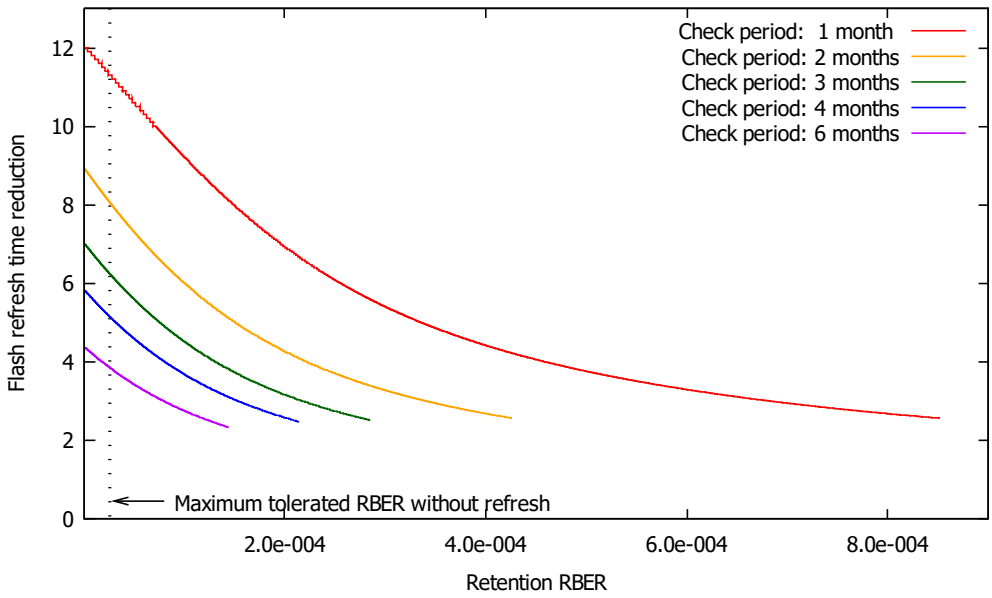

Figure 4.14: Access time reductions with respect to a systematic refresh scheme with a fixed refresh frequency. Flash memory pages with 16 Kb vulnerable bits and an ECC able to correct up to 10 bits

page, refresh operation are almost systematic and only avoided in the absence of retention errors. Here, the linear approach is applied to Flash memories and SSDs where only a small number of units is prone to errors. Thus, the average access time reduction is expected to approach the maximum values that correspond to retention RBER values managed by the available ECC only.

4.4 Conclusion

In this chapter, a statistical approach was proposed to improve the reliability of MLC NAND Flash memories in enterprise-class SSD. Reliability improvement is based on the increase of the tolerated retention RBER calculated by means of left retention time estimation. This estimation is done at periodic check operations of memory pages, and also at functional reads. Left retention time calculation is based on the detected number of retention errors and the retention age of data during each check or functional read. Left retention time is estimated in relation to a target value of UBER and a target overall retention time. Refresh operations are performed at check or read operations if the left retention time does not guarantee the integrity of data till the next check operation. This method allows to improve retention RBER by up to 28x for a check period of 1 month and a 3 years overall target retention time. In addition, the average time between refresh operations with this method is adapted to the actual retention RBER of the memory pages.

The statistical approach was then simplified by the consideration of a linear approximation

of retention RBER. This approximation allows to simply calculate the left retention time based on the retention errors number and retention age, without the statistical estimation of the failure rate λ. The tolerated retention RBER is also improved with this method when the upper limit of UBER is respected. Similarly to the statistical approach, functional reads can be used to perform check operations. This allows to better adapt the average time between refresh operations to the actual retention RBER of the memory pages. The average time between refresh operations increases. Therefore, access time of Flash memories for refresh operations is reduced compared to a systematic refresh scheme. Besides, this method allows to improve retention RBER by up to 35x for a check period of 1 month and a 3 years overall target retention time.

Chapter 5

Temperature influence on SEU vulnerability of SRAM cells

For the second part of our work, we focus on temperature effects on the chosen volatile memories. In this chapter, we study the temperature influence on SEU vulnerability of standard 6T-SRAM cells and of hardened DICE. This is done by estimating of the critical charge, i.e. the minimal collected charge capable of altering the target cell logic state. The critical charge is related to the current pulse induced by the collection of the electric charges due to the presence of electric fields in active areas of transistors. In the first section of this chapter, we start by exposing the problem of SRAM leakage during the hold phase. Previous studies on the influence of temperature on Soft Error Rate (SER) in SRAM are then presented and followed by an overview of existing models of the current pulse generated by ionizing particles. In the second section, we give some details on the SPICE simulations that were done to evaluate the critical charge of 6T-SRAM and DICE for different technologies and temperatures. The last section present the analysis of the obtained results and conclusions.

5.1 Leakage in SRAM and temperature influence on sensitivity to radiations

For safety and critical applications, terrestrial radiation resulting from cosmic rays is one of the main concerns. This is particularly true for circuits intended for avionics and military use. To evaluate the sensitivity to radiation effects of such circuits with new technology nodes, experiments and simulations are done. SRAM are usually used as test elements for such circuits. In fact, due to their feedback loop, they are particularly sensitive to soft errors and they are able to record the upsets. In addition, they are widely used in integrated circuits and system-on-chip as memory elements due to their speed and compatibility with standard logic processes. This makes them a major source of soft errors. Technology scaling brought important SRAM cell size reduction. However, as the cell area and supply voltage need to be kept as low as possible in order to limit leakage power consumption and improve integration density, SRAM cells vulnerability to soft errors increases.

In this section, we focus on SRAM leakage during the hold phase to justify our choice of considering the data retention phase in SRAM memories to study the combined effect of temperature and radiation. Previous studies of the temperature influence on SER in SRAM are also considered with emphasis on the used models of current pulses generated by ionizing particles.

5.1.1 SRAM leakage in hold phase

During the hold phase in 6T-SRAM cells, the pair of inverters is intended to hold the stored value as long as the supply voltage is applied. The two bitlines are disconnected from the internal nodes of the SRAM (Q and QB Fig. 1.16) that have complementary logic states. Due to the cross-coupled routing of the pair of inverters and depending on the stored value, two of the internal transistors are ON (P1 and D2 or P2 and D1 Fig. 1.16) and the two others are OFF (P2 and D1 or P1 and D2, respectively Fig. 1.16). For OFF-state transistors, the flow of parasitic electrons in the channel results in subthreshold leakage current I_{sub}. According to (1.12), the temperature dependence of I_{sub} presents a threat for the reliability of 6T-SRAM cell [27]. With temperature rise, subthreshold leakage increases in SRAM cells [80]. This is in part due to the decrease of the transistor threshold voltage with temperature (section 1.2.1).

Several studies are conducted in order to find adequate solutions for minimizing SRAM leakage in hold state. One of the tendencies consists in reducing the standby supply voltage to its limit before data retention problems occurrence. This value is called Data Retention Voltage (DRV) and it substantially reduces leakage and power consumption in 6T-SRAM [81][82]. Nevertheless, the reduction of supply voltage in hold state results in the opposite effect on the reliability of SRAM cells. In SRAM cells, radiation sensitivity problems are exacerbated in the case of supply voltage decrease.

For all these reasons, we chose to study the combined effects of temperature and radiation on SRAM cells in the standby phase. Charges accumulation and collection due to ionizing particles result in a current pulse in OFF-state transistors in SRAM cells. When added to the subthreshold leakage current, this current pulse can eventually trigger in a bit-flip. As leakage current is highly dependent on temperature, thermal variations have an influence on Single Event Upsets (SEU) sensitivity, which justifies our interest in studying their impact on standard 6T-SRAM cells and hardened DICE.

5.1.2 Temperature effect on soft error rate in SRAM

Different studies have been conducted to highlight the effect of temperature on the vulnerability of SRAM cells to radiations. Many of them showed the interest of the research community in the evaluation of Soft Error Rate (SER) of SRAM under different operating conditions that include supply voltage, process and temperature variations. This has been done for atmospheric neutron radiation by experimental use of atmospheric-like neutron beam [83] and by accurate simulations of atmospheric neutron-induced currents [84]. Similar studies have been done for an SRAM with a particular technological node (commercial 90 nm SRAM in [83]). In [85], temperature impact is evaluated for commercial SRAM from different vendors. Other studies have been conducted by the analysis of heavy ion-induced transient current model using TCAD simulations [86][87]. Temperature effect on SEU sensitivity is given for specific technological nodes, 180 nm SRAM in [86] and 90 nm SRAM in [87]. In [88], temperature impact on heavy ion-induced SEU vulnerability

is evaluated through experimental cross section measurements.

Our study on the effect of temperature on SEU vulnerability takes place in this context. Our approach is based on the analysis of the temperature effect on the minimum collected charge that can result in an SEU, i.e. the critical charge (Q_{crit}), using a specific model for the current pulse which is generated by the ionizing particle. Our study is an extension to previous work concerning the temperature impact on soft error rate of conventional 6T-SRAM. In fact, the evaluation of Q_{crit} is done for both standard 6T-SRAM cells and for DICE, which are hardened cells that have an improved resilience to radiation-induced SEU. A comparison of the impact of temperature is done for the two types of cells and for different technological nodes through SPICE simulations.

5.1.3 Models of transient current pulses generated by ionizing particles

According to section 5.1.1, in standard 6T-SRAM cells, nodes sensitive to radiations are the drains of OFF-NMOS and OFF-PMOS transistors in the inverters pair. The interaction of an ionizing particle with a sensitive node provokes a current transient and can result in flipping the stored value. SEU are then modeled by the injection of a current pulse at the chosen sensitive node. Different models of transient current pulses have been used for the studies of SEU occurrence in SRAM.

In [15], the transient current created by $\alpha-$particles is given by the following equation:

$$I_{pulse}(t) = \frac{2}{a_t\sqrt{\pi}}\sqrt{\frac{t}{a_t}}e^{-\frac{t}{a_t}} \tag{5.1}$$

where t is time, a_t is a time constant depending on different factors such as material and doping. The shape of this current pulse, shown in Fig. 5.1, depends on drift and diffusion currents contributions. The drift component gives the most rapid change rate in the current pulse, as opposed to the diffusion component which is responsible for the slower decay [15].

One of the well known and most used models is called *the double exponential model* [89][90][91][92]. It models a transient current pulse with a rapid rise time (t_r) and a gradual fall time (t_f) (Fig. 5.2) for which the shape is approximated by the following equation:

$$I_{pulse} = \frac{Q_{coll}}{t_f - t_r}(e^{-\frac{t}{t_f}} - e^{-\frac{t}{t_r}}) \tag{5.2}$$

where Q_{coll} is the collected charge due to the particle strike, t_r is the rise time and t_f the decay time.

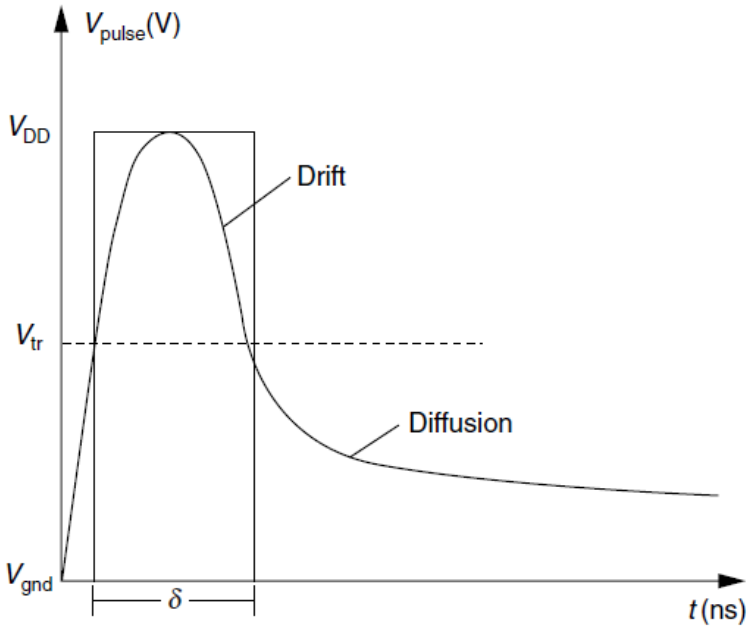

Figure 5.1: Transient current pulse shape caused by an α-particle strike with drift and diffusion contributions [15]

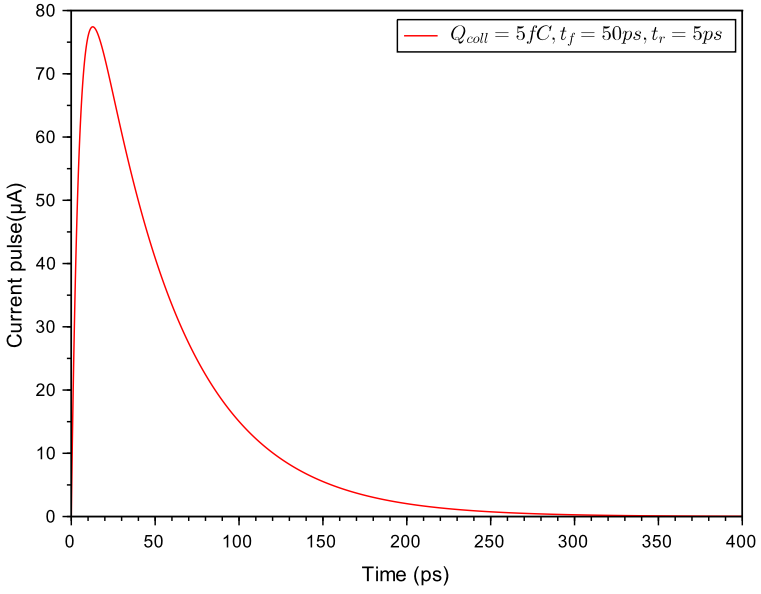

Figure 5.2: Example of a double exponential transient current pulse

According to [91][92] and using the derivative of the double exponential law, t_{max} the time at which the transient current reaches its maximum value I_{max} is given by the formula below:

$$t_{max} = \frac{\ln \frac{t_f}{t_r}}{\frac{1}{t_f} - \frac{1}{t_r}} \tag{5.3}$$

In addition, I_{max} is related to t_{max} by this equation:

$$I_{max} = \frac{Q_{coll}}{t_f - t_r}(e^{-\frac{t_{max}}{t_f}} - e^{-\frac{t_{max}}{t_r}}) \tag{5.4}$$

The double exponential law is characterized by a rapid rise and a slower decay. Thus, t_f and t_r are related by $t_f = kt_t$ with k being a constant greater than one [91][92]. In this case, (5.2) can be expressed as follows:

$$\begin{aligned} I_{pulse} &= \frac{kQ_{coll}}{(k-1)t_f}(e^{-\frac{t}{t_f}} - e^{-\frac{kt}{t_f}}) \\ &= \alpha I_{max}(e^{-\frac{t}{t_f}} - e^{-\frac{kt}{t_f}}) \end{aligned} \tag{5.5}$$

with α being a known constant resulting from the proportionality ratio k existing between t_f and t_r.

The critical charge is quantified by the transient current model used to determine, through electrical simulations, how a given memory cell flips for different shapes and intensities of the current pulse. In fact, particle strikes result in current transients with different pulse widths. Besides, the deposited charge in transistors sensitive nodes depends on the current pulse shape [93]. Thus, the use of an accurate model for current transients is essential to estimate Q_{crit} for the prediction of soft errors.

5.2 6T-SRAM and DICE designs

In this section, we describe the standard 6T-SRAM cell and DICE designs used for our simulations. For the purpose of this study, the two cells were implemented in a SPICE simulator as detailed below in this chapter. Details on the test structures and the dimensioning of transistors are given in this section.

5.2.1 Structure of the simulated 6T-SRAM cells

Here, the simulated 6T-SRAM cells are considered without the peripheral circuitry needed for read and write operations. The basic structure of a 6T-SRAM cell, given by Fig. 1.16, shows a

pair of cross-connected inverters with two NMOS access transistors. For the considered simulations, four resistors and two capacitances are added. The resistors are included in the two inverters cross-coupled structure to simulate the internal resistance on VDD and GND nodes. The two capacitances are connected to the access transistors and represent bitlines (BIT and BITB) equivalent capacitances (Fig. 5.3).

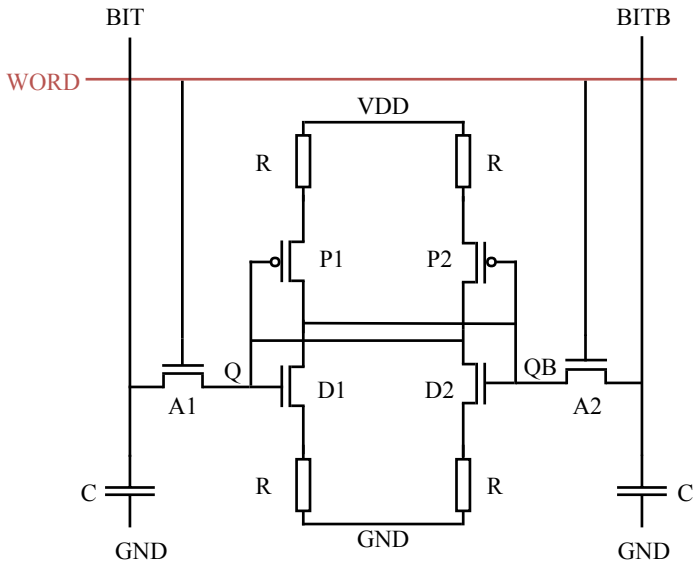

Figure 5.3: Schematic of the 6T-SRAM cell used in simulations

The 6T-SRAM cell allows the storage of a single bit, either a logic 1 or 0. In our study, we analyze the response of cells to radiation during retention mode. For the simulations, we can consider the storage of a logic 0 (node Q is at GND and node QB is at VDD) or a logic 1 (node Q is at VDD and node QB is at GND). This choice does not impact on simulation results due to cell symmetry. In retention mode, WORD is at logic 0 (GND) which forces the OFF state of access transistors and disconnects BIT and BITB respectively from Q and QB nodes to preserve the stored value. BIT and BITB are connected to VDD, which is the value commonly forced by the pre-charge circuit during retention mode.

5.2.2 Structure of the simulated DICE

As explained in section 2.2.2, a Dual Interlocked Storage Cell (DICE) is a storage CMOS cell with a special design able to increase Q_{crit} as compared to a standard 6T-SRAM cell. Thus, it has an improved resilience to radiation-induced SEU. Compared to a 6T-SRAM cell, a DICE relies on double the number of transistors and a wiring enabling each cell node to control at most one of the transistors that drive another node as it is shown in Fig. 2.13 [12][64][94]. For the considered

simulations, eight resistors and 4 capacitances have been added to the DICE in the same way as in the simulated 6T-SRAM cell (Fig.5.4).

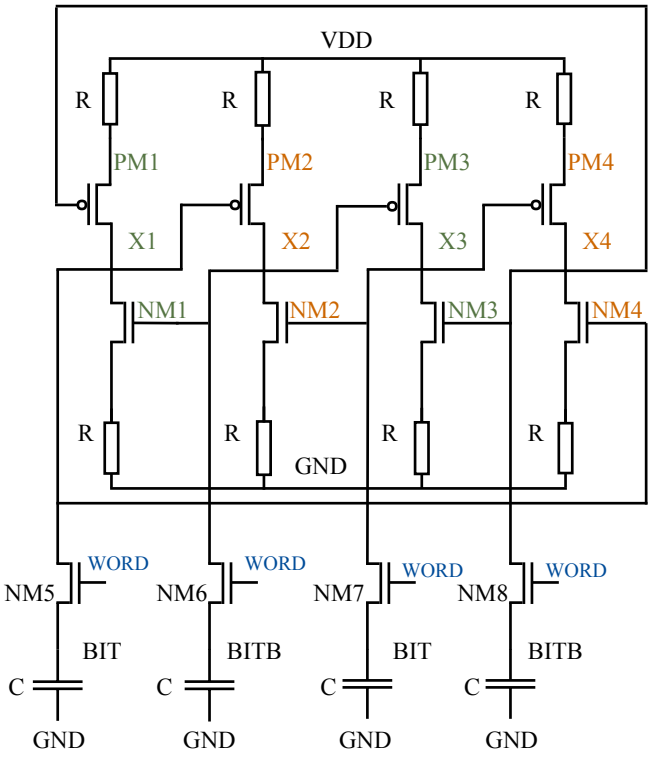

Figure 5.4: Schematic of the DICE used in simulations

Data storage nodes are organized into two complementary pairs (X1-X3 and X2-X4) and the four access transistors (NM5-NM6-NM7-NM8) are accessed during read and write operations. As we analyze the response of the DICE to radiation during retention mode, we chose to simulate the storage of a logic 0. The node X1 and X3 are at logic 0 (GND) and nodes X2 and X4 are at logic 1 (VDD). In retention mode and in the same way as in a 6T-SRAM cell, WORD is at logic 0 and BIT and BITB are connected to VDD and disconnected from the storage nodes (X1-X2-X3-X4). The same study could have been conducted for a logic 1 stored in nodes X1 and X3. With the DICE features of redundancy and feedback, the state of a sensitive storage node can be regenerated once it flips under the effect of single-event strikes. In fact, uncorrupted sections deliver the correct state restoring the feedback to recover the state of the affected node.

5.2.3 Transistors dimensioning in the simulated cells

As expressed in section 1.5.2, the dimensioning of transistors is an important matter in 6T-

SRAM cell design, since its stability and resilience to noise highly depends on it. The dimensions of transistors, especially in the cross-connected inverters, must respect several requirements:

- Integration limits reflected by pushing towards the use of minimal sizes allowed by technology nodes.
- Nominal symmetry need to ensure equal strength of stored logic 1 and logic 0.
- Enhancement of resilience to electromagnetic interference.
- *Read stability* that ensures the design of a cell sufficiently strong to be accessed during read operation without destroying its content.
- *Writability* that ensures the design of a cell not excessively strong in order to allow the change of the stored value during the write operation.

These requirements have been respected in our simulations of 6T-SRAM cells. We chose to conduct our study with four technology nodes: 65 nm, 45 nm, 32 nm and 22 nm. It has to be noted that the dimensioning of the different transistors in the 6T-SRAM cells has been done in adequacy with embedded 65 nm SRAM dimensioning in industry. However, the transistors dimensions have been scaled in order to use 65 nm as a minimum length for the transistors in the two inverters loop. For 45 nm, 32 nm and 22 nm technology nodes, transistors dimensions have been extrapolated from those of the 65 nm transistors using SNM calculation as a method of validation and the technology node feature size as the minimum gate length. The dimensions used in 6T-SRAM simulations are depicted in Table 5.1.

Dimensions (nm)	65 nm		45 nm		32 nm		22 nm	
	W	L	W	L	W	L	W	L
PMOS transistors	87	65	60	45	43	32	29	22
NMOS transistors	125	65	86	45	61	32	42	22
Access transistors	92	76	64	52	45	37	31	26

Table 5.1: Transistors dimensions for 6T-SRAM cells

For each designed cell, the SNM is evaluated in a large range of temperatures (from $-50\,°C$ to $150\,°C$). In the retention mode, the SNM is the maximum value of noise that can be tolerated by the cell without flipping the stored value. It is calculated through the *butterfly curve* [41][95]. As we chose the same dimensions for NMOS and PMOS transistors in both inverters, the *butterfly curve* is symmetric and its two loops are identical. Thus, SNM can be deduced from the side length of one of the maximum squares that fit in the loops.

DICE resilience performance does not depend on optimal transistor sizing [64]. Thus, for DICE simulations, we used the same transistor sizes as in the 6T-SRAM cell for the four chosen technology nodes. This allows a fair comparison between the two types of cells. In addition, we didn't produce any *butterfly curves* for the DICE. In fact, the electrical scheme of this cell results in a different definition of its SNM calculations which is out of the scope of our study.

5.3 Simulation setup

The different simulations for SNM calculation and Q_{crit} estimation for both 6T-SRAM and DICE have been conducted using LTspice XVII simulation software with both schematics and netlists. In addition, a python script has been used to iterate Q_{crit} estimation for the considered technology nodes in the temperature range $-50\,°C$ to $150\,°C$. SPICE models of bulk CMOS transistors extracted from the predictive technology models (PTM) have been used in our simulations for both 6T-SRAM and DICE [96]. The considered supply voltages correspond to the PTM transistors predicted values for the different technology nodes [97]. They are given in Table 5.2.

Technology node (nm)	65	45	32	22
Supply voltage (V)	1	1	0.9	0.8

Table 5.2: Supply voltages for 6T-SRAM and DICE simulations

For SNM calculations, a first set of simulations has been performed with 6T-SRAM cells only. In order to draw the *butterfly curves*, nodes Q and QB have been initially set to logic 1 and logic 0, respectively. A voltage source has been added to simulate the effect of noise on nodes Q and QB alternatively. Q is drawn versus QB and QB versus Q simultaneously which gives the *butterfly curve* as shown in Fig. 5.5.

The value of SNM is extracted by a graphical technique detailed in [95]. The idea is to use a coordinate system (u,v) rotated by 45° with respect to the (x,y) coordinate system showing Q versus QB and QB versus Q as shown in Fig. 5.5. In the (u,v) coordinate system, the subtraction of the two v values of normal and mirrored inverter characteristics (v1 and v2 respectively) at a given u value, gives the measure of the diagonal length of the squares that can be fitted in the loops of the butterfly curves. Therefore, the multiplication by $\frac{1}{\sqrt{2}}$ of the maximum $|V1 - V2|$ gives the value of SNM [95].

$$SNM = \frac{1}{\sqrt{2}} max|V1 - V2| \tag{5.6}$$

The obtained SNM values are given in Table 5.3. This table shows that the transistor sizes, in the 6T-SRAM cells for the different technology nodes, have been chosen in order to have a ratio of

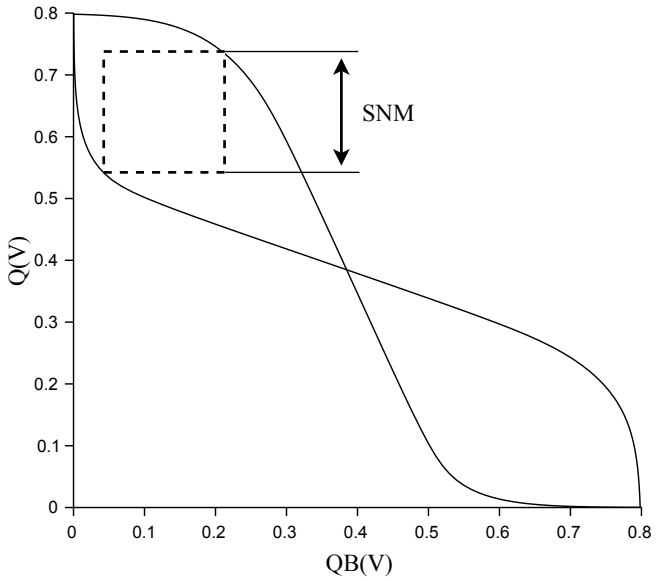

Figure 5.5: Butterfly curve and SNM calculation for 22 nm 6T-SRAM cell

hold SNM over supply voltage around 0.3, which is chosen as an acceptable value in the case of our simulations.

Technology node	Temperature (°C)			
	-50	27	90	150
65 nm	0.344	0.314	0.295	0.279
45 nm	0.328	0.296	0.276	0.258
32 nm	0.281	0.251	0.231	0.214
22 nm	0.224	0.182	0.156	0.137

Table 5.3: SNM evolution with temperature for 6T-SRAM cells

For Q_{crit} estimation, we consider the 6T-SRAM and DICE designs of Fig. 5.3 and Fig. 5.4. We consider the case of the storage of a logic 0 for both cells. Here, the nodes connected to the drains of NMOS and PMOS transistors in OFF states are the most sensitive to particle strike. The amount of collected charge in these sensitive nodes depends on the shape and duration of the current pulse resulting from the interaction of a particle. To simulate the effect of a particle strike, a current source is added in parallel to the considered OFF transistor. Here, we chose to the apply the current sources on the OFF-NMOS transistors for both 6T-SRAM and DICE as it has been

done in previous studies [40]. Considering the storage of a logic 0, the corresponding transistors are D1 in 6T-SRAM cell (Fig. 5.3) and NM2 (or NM4) transistor in DICE (Fig. 5.4). We chose to consider the double exponential law for the transient current model (5.2). In addition, we defined initial values of time constants t_f and t_r, and also the initial value of the maximum current I_{max}. As explained in [91][92] and according to (5.5), we chose a decay time 50 times higher than the rise time. This results in the following equation:

$$I_{pulse} = \alpha I_{max}(e^{-\frac{t}{t_f}} - e^{-\frac{50t}{t_f}}) \tag{5.7}$$

with α a known constant resulting from the proportionality constant $k = 50$.

As fixed initial values, we chose:

- $t_r = 10$ ps
- $t_f = 500$ ps
- $I_{max} = 100\ \mu A$

It should be noted that these values have been chosen in agreement with those found in the bibliography. The challenge was to fix the order of magnitude of these values in adequacy with existing diffusion and transient current models and in a way that allows a bit-flip in both 6T-SRAM and DICE even though they don't have the same resilience to radiation [15][90][91][92]. For this purpose and in order to evaluate Q_{crit} for the considered technology nodes and temperatures, the double exponential current pulse is modulated via a proportionality constant β that affects both time constants and I_{max} as shown in (5.8):

$$I_{pulse} = \alpha\beta I_{max}(e^{-\frac{t}{\beta t_f}} - e^{-\frac{50t}{\beta t_f}}) \tag{5.8}$$

As a consequence, the resulting current pulse keeps the same shape but the induced charge depends on the β value. With a higher value of β, the current pulse is wider and the amplitude of the current peak is bigger, which results in an increased deposited charge. The critical charge is deduced from the transient current pulse I_{pulse_min} with the smallest value of β that results in a flip of the stored value as in the case of an SEU. Q_{crit} is the integral of I_{pulse_min} in the time domain. Fig. 5.6 shows examples of minimal current pulses able to flip a 32 nm DICE at two different temperatures.

5.4 Simulation results and analysis

In this section, we present the results of Q_{crit} estimation for 6T-SRAM and DICE cells implemented in 65 nm, 45 nm, 32 nm and 22 nm technology nodes and at storage temperatures ranging

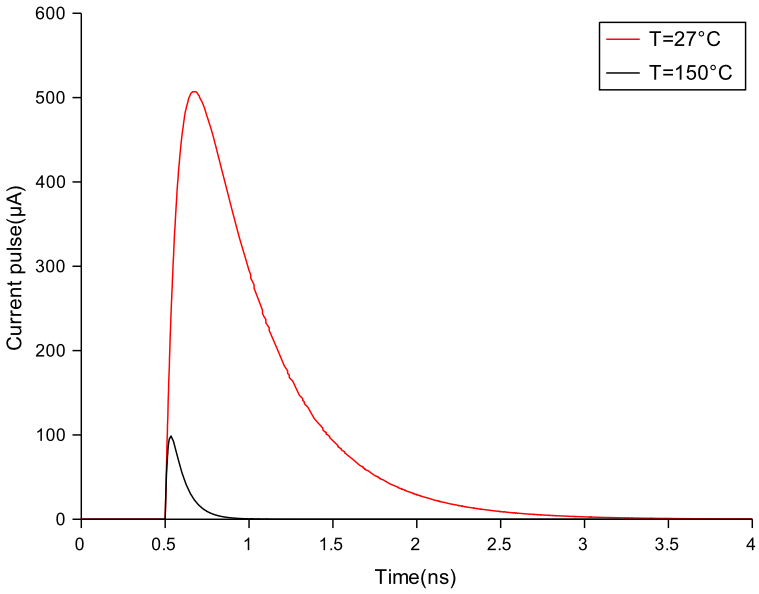

Figure 5.6: Examples of minimal current pulses able to flip a 32 nm DICE

from −50 °C to 150 °C. Simulation results are analyzed and conclusions are given at the end of this section.

5.4.1 6T-SRAM cells results

The evolution of Q_{crit} in the chosen temperature range and for the different technology nodes is given in Fig. 5.7.

Here, Q_{crit} has an order of magnitude that varies between 10 fC at −50 °C and 10^{-1} fC at 150 °C. The critical charge monotonically decreases with the increase of temperature and with the reduction of the technology node. This is consistent with the evolution of electrical parameters with temperature and technology scaling. In fact, electrical parameters are degraded with temperature. As explained in section 1.2.1, the drain current of the transistors in ON state (I_{on}) decreases under the effect of temperature. On the other hand, the subthreshold leakage current I_{sub} increases as explained in section 5.1. In this case, I_{on} can be overcome by weaker current pulses in the OFF-state transistors that drive the same node, which makes it easier for the 6T-SRAM cell to undergo a bit-flip. This is reflected by the decrease of Q_{crit} with temperature. With technology shrinking, the equivalent capacitances of sensitive nodes are reduced. As the first approximation of the critical charge is the product of threshold voltage by capacity of sensitive nodes, this results in the decrease of Q_{crit} for smaller technology nodes.

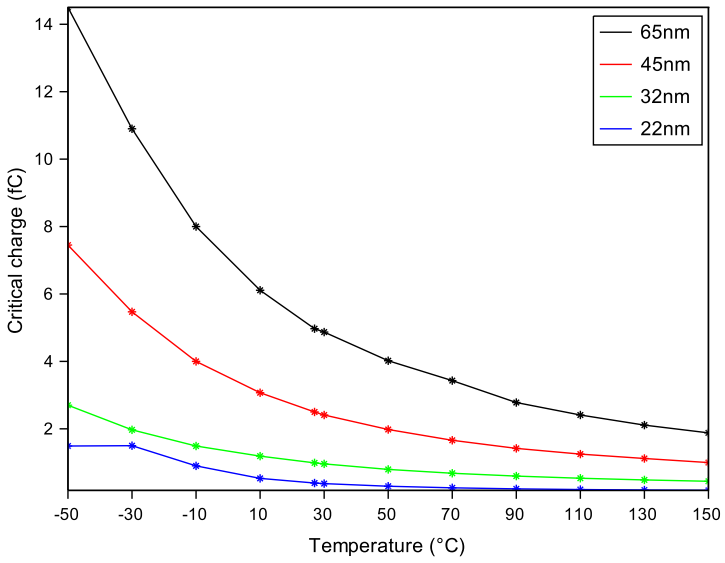

Figure 5.7: Critical charge evolution with temperature for 6T-SRAM cells

Fig. 5.8 shows the ratio of Q_{crit} at −50 °C, 90 °C and 150 °C with respect to a nominal temperature of 27 °C. For all technology nodes, the maximum ratio is obtained for the minimum temperature of the considered range (−50 °C). In addition, over the whole range of temperatures, Q_{crit} is increased by up to 88.4% as compared to its minimal value at 150 °C.

5.4.2 DICE results

The evolution of Q_{crit} in the chosen temperature range and for the different technology nodes is given in Fig. 5.9.

The order of magnitude of Q_{crit} varies between 10^7 fC at −50 °C and 1 fC at 150 °C. As for 6T-SRAM cells, Q_{crit} of DICE cells decreases monotonically with temperature and technology shrinking. The electrical parameters are degraded and the capacitance of sensitive nodes decrease with temperature and technology shrinking in the same way as for 6T-SRAM. Fig. 5.10 shows the ratio of Q_{crit} at −50 °C, −10 °C and 10 °C with respect to a nominal temperature of 27 °C. The maximum Q_{crit} ratio is obtained for the minimum temperature of the considered range (−50 °C). Over the whole range of temperature, Q_{crit} is increased by up to 99.9% as compared to its minimal value at 150 °C.

The comparison of Q_{crit} in both 6T-SRAM and DICE cells results in the following observations that are in adequacy with the higher resilience to SEU observed in DICE compared to 6T-SRAM cells:

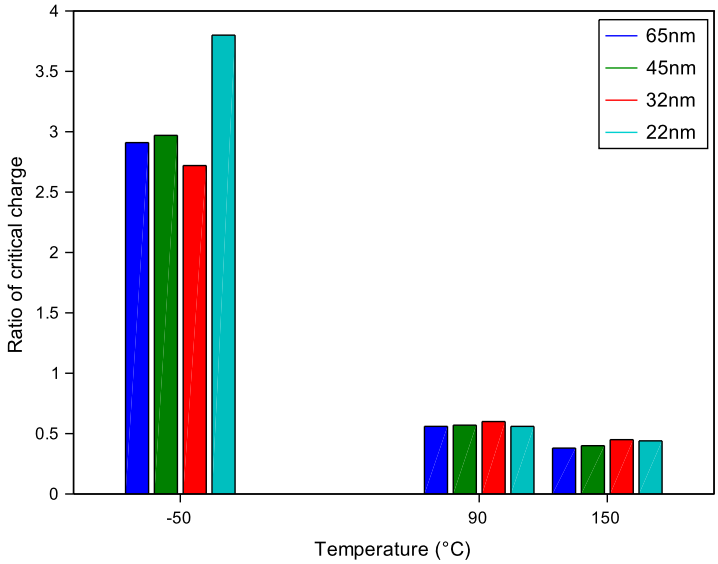

Figure 5.8: Critical charge ratio compared to 27 °C for 6T-SRAM

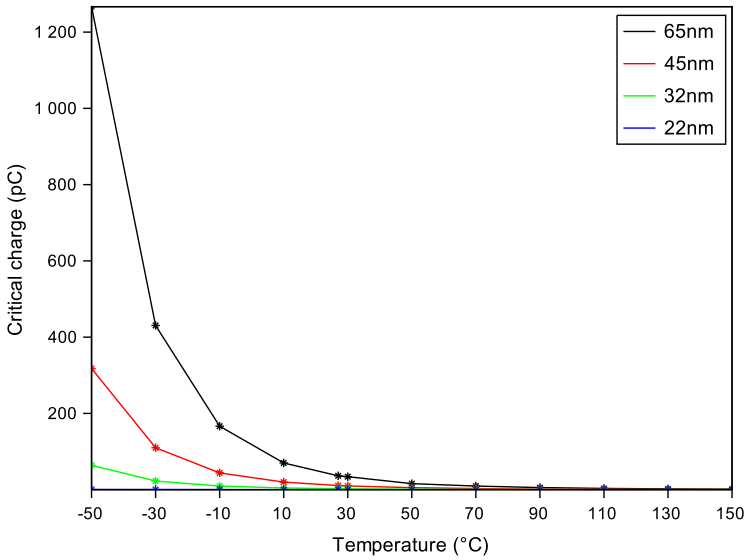

Figure 5.9: Critical charge evolution with temperature for DICE

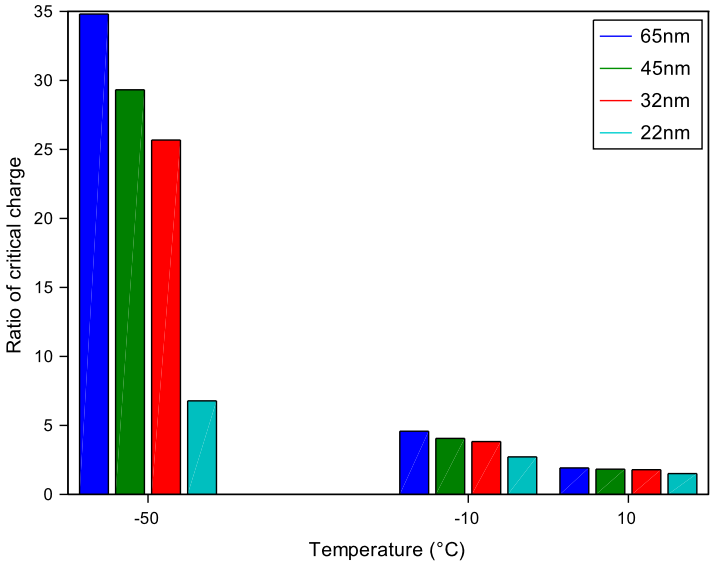

Figure 5.10: Critical charge ratio compared to 27 °C for DICE

- For the −50 °C to 150 °C temperature range, Q_{crit} in 6T-SRAM varies between 0.17 fC and 14.5 fC.
- For the same temperature range, Q_{crit} in DICE varies between 9.6 fC and 1270 pC.
- The maximum ratio of Q_{crit} between 6T-SRAM and DICE cells is equal to 8.7x10^4. It corresponds to 65 nm cells at −50 °C.
- Q_{crit} of 22 nm DICE at 130 °C and 150 °C has the same order of magnitude as Q_{crit} of 6T-SRAM cells at −30 °C and −50 °C.

5.5 Conclusion

The reliability of CMOS storage cells is highly dependent on temperature. Leakage and radiation sensitivity problems are exacerbated in case of temperature increase for both standard SRAMs and hardened cells. Here, we studied the thermal variations influence on Single Event Upsets (SEU) sensitivity of 6T-SRAM and DICE cells, in correlation with technology scaling.

SPICE simulations, using bulk CMOS transistors from the predictive technology models for 65 nm, 45 nm, 32 nm and 22 nm technology nodes, were conducted for both types of cells. SNM calculations were carried out to ensure the adequate dimensioning and stability of the designed 6T-SRAM cells. The same transistors dimensions were used for DICE designs to allow a fair comparison

between the two types of storage cells. A double exponential model was retained for the simulation of the current induced by ionizing particle strikes. The resulting values of critical charge (Q_{crit}) for SEU triggering over a temperature range between $-50\,°C$ and $150\,°C$ were measured.

Simulations showed that Q_{crit} is sensibly reduced by the rise of temperature and monotonically decreases with technology scaling. Over the whole temperature range, Q_{crit} variations of up to 88.4% and 99.9% have been simulated for 6T-SRAM and DICE cells, respectively, which is in adequacy with the higher resilience to SEU observed in DICE compared to 6T-SRAM cells.

General conclusion

The complexity of microelectronic systems is in constant increase due to very large scale integration on chips and the continual scaling of transistors. In addition, the increasing demand for higher performance and lower power consumption pushes toward the limits of the existing technologies and processes. In this context, reliability, which characterizes the ability of a system or device to operate correctly under specific environmental conditions, becomes a concern. Reliability problems are exacerbated in harsh environments where extreme conditions alter the proper functioning of systems and devices. For different industry players, high temperature exposure is considered as one of the main reliability concerns. Data storage components are highly affected by temperature variations and repetitive thermal cycles. As memories represent a great majority of embedded devices in systems on chip, the reliability of volatile and non-volatile mediums should be guaranteed through an entire range of operating temperature adapted to the considered application.

My work evolved around the study of reliability and temperature effects on two types of memories: NAND Flash memories and SRAMs. The two main targets of my thesis were (a) the proposition of monitoring techniques for NAND Flash memories in order to improve their retention and extend their lifetime and (b) the study of temperature effects on the resilience to single event upsets in a standard SRAM in comparison to a hardened cell.

A first part of the work consisted in the investigation of the mutual dependence between retention time and cycling endurance in Flash memories. A timer-based solution was proposed to reduce the number of refresh operations with respect to a pessimistic refresh scheme in MLC NAND Flash used in enterprise-class SSDs. The timer enables the integration of the effect of temperature variations on the memories retention time. In fact, it allows to efficiently approximate the Arrhenius law that gives the retention time evolution with temperature in MLC NAND Flash memories. The operating temperature range is then divided into sub-intervals by the means of the timer. Each sub-interval has a constant approximated retention time. The refresh frequency is then adapted the temperature variations and is highly reduced compared to a fixed frequency refresh scheme.

The second phase was dedicated to the improvement of MLC NAND Flash memories reliability in enterprise-class SSDs. Reliability improvement was based on a statistical approach that allows the increase of the tolerated raw bit error rate in NAND Flash memories via the statistical estimation of the reliable left retention time. This estimation is done at periodic check operations and can result in a refresh if the left retention time does not guarantee the integrity of data until the next check operation. The functional reads of memory pages can be used to execute the check operations of the proposed scheme. In this case, a conventional timer, or a temperature-aware timer as the one proposed in the above, must be used to further reduce the number of refresh operations. A computational approximation of this statistical approach was also proposed. The average time between refreshes was estimated for both the statistical and approximated scheme. The reduction of refresh access time was also simulated for the approximation scheme to prove the efficiency of this monitoring technique.

In the last part, SRAM reliability was investigated under the effect of temperature. The thermal variations influence on single event upsets was addressed in the case of standard 6T-SRAM cells and hardened DICE cells. SPICE Simulations were conducted in order to validate the designs and dimensions of the cells and also to choose a model to simulate the current induced by ionizing particles strike. The values of critical charge for SEU triggering were measured for both types of cells over a large temperature range (from $-50\,°C$ to $150\,°C$). The simulations were also conducted for both cell designs with different technology nodes (65 nm, 45 nm, 32 nm and 22 nm). Finally, this critical charge study gave compliant results with the higher resilience to SEU observed in DICE compared to 6T-SRAM cells. The maximum ratio of Q_{crit} between 6T-SRAM and DICE cells was found equal to $8.7\text{x}10^4$.

Appendix A

UBER estimation in case of absence of check operations

When no check and refresh operations occur by retention time t_{age}, UBER can be simply computed using (2.3). Here, we assume that retention errors are predominant over disturb errors in the considered Flash memories [57][56][58] and that retention errors are the only ones able to accumulate in time [14]. UBER in the considered memory page is calculated as follows:

$$UBER(t_{age}) = \frac{1}{N} \sum_{i=M-N_{non\_ret}+1}^{N_{vul}} \binom{N_{vul}}{i} (RBER_{ret}(t_{age}))^i (1 - RBER_{ret}(t_{age}))^{N_{vul}-i} \quad \text{(A.1)}$$

Where:

- N is the number of bits of the considered Flash memory page
- M is the maximum number of errors correctable by the ECC in the Flash page
- N_{vul} is the number of bits vulnerable to retention errors in the Flash page
- N_{ret} and N_{non_ret} are respectively the number of bits affected by retention and disturb errors in the Flash page
- t_{age} is the retention age of data in the Flash page
- $RBER_{ret}(t_{age})$ is the retention RBER over t_{age} calculated according to (4.11) for the statistical estimation method and according to (4.12) for the linear approximation method

Appendix B

UBER estimation in case of check operations

When periodic check operations are performed according to the schemes of algorithm 1 or 2, refresh operations may take place if the left retention time does not guarantee the integrity of data till the following check. UBER can be calculated as the sum of the contributions to UBER during check periods t_{check} and over the target retention time t_{max} (3 years in our study).

$$UBER = \frac{1}{N} \sum_{i=1}^{max} UBER(i) \tag{B.1}$$

Where:

- N is the number of bits of the considered Flash memory page
- $max = ceiling(\frac{t_{max}}{t_{check}})$ is the smallest integer greater than or equal to $\frac{t_{max}}{t_{check}}$
- UBER(i) is the contribution to overall UBER of uncorrectable errors that may appear during t_{check} that separates check operations occurring at $(i-1)t_{check}$ and it_{check}

At each check operation, a read is performed. When the *if condition* of algorithms 1 or 2 are not verified, no refresh occurs and errors are only handled by the ECC. UBER(i) is then estimated as the sum of the probabilities of (a) retention errors occurrence at the $(i-1)^{th}$ check operation that does not result in a refresh and (b) and retention errors occurrence at the i^{th} check operation that exceed the correction capacity of the ECC.

$$UBER(i) = \sum_{N_{ret}=0}^{N_{i-1}} P((i-1)t_{check}, N_{vul}, N_{ret}) \left[1 - \sum_{N'_{ret}=0}^{M-N_{ret}-N_{non\_ret}} P(t_{check}, N_{vul} - N_{ret}, N'_{ret}) \right] \tag{B.2}$$

Where:

- $P((i-1)t_{check}, N_{vul}, N_{ret})$ is the probability of N_{ret} occurrence by the time $(i-1)t_{check}$ and that does not result in a refresh at the $(i-1)^{th}$ check operation according to algorithms 1 or 2

- N_{i-1} is the maximum number of retention errors that can be withstood by the time $(i-1)t_{check}$ and that does not result in a refresh at the $(i-1)^{th}$ check operation according to algorithms 1 or 2
- $P(t_{check}, N_{vul} - N_{ret}, N'_{ret})$ is the probability of N'_{ret} occurrence during a check period t_{check} and that can still be handled by the available ECC at the i^{th} check
- M is the maximum number of errors correctable by the ECC in the Flash page
- N_{vul} is the number of bits vulnerable to retention errors in the Flash page
- N_{ret} and N'_{ret} are the tolerated number of bits that can affected by retention errors in the Flash page
- N_{non_ret} is the number of non-retention errors in the Flash page

The probability of N_{ret} occurrence by the time it_{check} is calculated as the sum of probabilities of the different distributions of retention errors over the storage time it_{check}. The use of N_{i-1} indicates that some scenarios are not possible and that, in this case, a check operation can result in a refresh at $(i-1)t_{check}$. This probability is calculated as follows:

$$P(it_{check}, N_{vul}, N_{ret}) = \sum_{N'_{ret}=0}^{min(N_{ret}, N_{i-1})} P((i-1)t_{check}, N_{vul}, N'_{ret}) P(t_{check}, N_{vul} - N'_{ret}, N_{ret} - N'_{ret}) \tag{B.3}$$

For i=1, this probability is calculated as follows:

$$P(t_{check}, N_{vul}, N_{ret}) = \binom{N_{vul}}{N_{ret}} (RBER_{ret}(t_{check}))^{N_{ret}} (1 - RBER_{ret}(t_{check}))^{N_{vul} - N_{ret}} \tag{B.4}$$

where $RBER_{ret}(t_{age})$ is the retention RBER over a check period calculated according to (4.11) for the statistical estimation method and according to (4.12) for the linear approximation method.

Bibliography relative to the study

Chapter 3:

- M. Seif, E. Farjallah, F. Badets, E. Chabchoub, C. Layer, J.-M. Armani, F. Joffre, C. Anghel, L. Dilillo, and V. Gherman, "Refresh frequency reduction of data stored in SSDs based on A-timer and timestamps," in 2017 22nd IEEE European Test Symposium (ETS), (Limassol, Cyprus), pp. 1–6, IEEE, May 2017.

Chapter 4:

- V. Gherman, E. Farjallah, J. Armani, M. Seif, and L. Dilillo, "Improvement of the tolerated raw bit error rate in NAND Flash-based SSDs with the help of embedded statistics," in 2017 IEEE International Test Conference (ITC), pp. 1–9, Oct 2017.

- E. Farjallah, J. Armani, L. Dilillo, and V. Gherman, "Improvement of the tolerated raw bit error rate in NAND Flash-based SSDs with selective refresh," in Elsevier Microelectronics Reliability Journal, pp. 1–9, 2018.

Chapter 5:

- E. Farjallah, V. Gherman, J.-M. Armani, and L. Dilillo, "Evaluation of the temperature influence on SEU vulnerability of DICE and 6T-SRAM cells," in 2018 13th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), (Taormina), pp. 1–5, IEEE, Apr. 2018.

Bibliography

[1] P. K. Lala, *Self-Checking and Fault Tolerance Digital Design*, morgan kaufmann publishers ed., San Francisco, 2001.

[2] N. H. E. Weste and D. M. Harris, *CMOS VLSI design: a circuits and systems perspective*, 4th ed. Boston: Addison Wesley, 2011.

[3] R. Micheloni, L. Crippa, and A. Marelli, *Inside NAND Flash Memories*. Dordrecht: Springer Netherlands, 2010.

[4] Swissbit, "X-500 / X-55 Series SLC vs. EM-MLC," Germany, Tech. Rep. WhitePaper_X-5xx_EM-MLC vs SLC_Rev100.doc, Apr. 2014.

[5] I. Micron Technology, "Nand flash 101: An introduction to nand flash and how to design it in to your next product," Technical note TN-29-19, 2006.

[6] James F. Salzman and Texas Instruments, "Total Ionizing Dose (TID) and Single Event Effects (SEE) Test Report," Test report, Dec. 2013.

[7] W. Paper, A. R. Olson, and D. J. Langlois, "Solid State Drives Data Reliability and Lifetime - Imation Corp," White paper, 2008.

[8] A. Spinelli, C. Compagnoni, and A. Lacaita, "Reliability of NAND Flash Memories: Planar Cells and Emerging Issues in 3d Devices," *Computers*, vol. 6, no. 2, p. 16, Apr. 2017.

[9] X. Wang, G. Dong, L. Pan, and R. Zhou, "Error Correction Codes and Signal Processing in Flash Memory," in *Flash Memories*, pp. 57–82.

[10] Y. Cai, O. Mutlu, E. F. Haratsch, and K. Mai, "Program interference in MLC NAND flash memory: Characterization, modeling, and mitigation." IEEE, Oct. 2013, pp. 123–130.

[11] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 305–316, Sep. 2005.

[12] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Transactions on Nuclear Science*, Dec. 1996.

[13] Y. Cai, G. Yalcin, O. Mutlu, and E. F. Haratsch, "Error Analysis and Retention-Aware Error Management for NAND Flash Memory," vol. 17, no. 1, p. 25, 2013.

[14] N. Mielke, T. Marquart, Ning Wu, J. Kessenich, H. Belgal, E. Schares, F. Trivedi, E. Goodness, and L. R. Nevill, "Bit error rate in NAND Flash memories." IEEE, Apr. 2008, pp. 9–19.

[15] E. Dupont, M. Nicolaidis, and P. Rohr, "Embedded robustness ips for transient-error-free ics," *IEEE Design Test of Computers*, vol. 19, no. 3, pp. 54–68, May 2002.

[16] Doug Kearns and Cypress, "Practical Guide to Endurance and Data Retention," Application note AN99121 - Rev. *B, Feb. 2017.

[17] I. roadmap committee, "International technology roadmap for semiconductor - process integration devices and structures," Tech. Rep., 2013.

[18] C. T. C. Automotive Electronics Council, "Failure mechanism based stress test qualification for integrated circuits," Tech. Rep. AEC - Q100 - Rev-G, May 2007.

[19] R. A. Normann, "First high-temperature electronics products survey 2005," Sandia National Laboratories, Tech. Rep., 2006.

[20] Alexander Muffler, X-Fab, "Challenges for Non Volatile Memory (NVM) for Automotive High Temperature Operating Conditions," Munich, Germany, Nov. 2017.

[21] R. Cojbasic and Y. Leblebici, "Design of high-temperature SRAM for reliable operation beyond 250°C," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. Lisbon, Portugal: IEEE, May 2015, pp. 2545–2548.

[22] R. Cojbasic, O. Cogal, P. Meinerzhagen, C. Senning, C. Slater, T. Maeder, A. Burg, and Y. Leblebici, "FireBird: PowerPC e200 based SoC for high temperature operation," in *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*. San Jose, CA, USA: IEEE, Sep. 2013, pp. 1–4.

[23] S. M. Sze and K. K. Ng, *Physics of semiconductor devices*, 3rd ed. Hoboken, N.J: Wiley-Interscience, 2007.

[24] D. Wolpert and P. Ampadu, "Temperature effects in semiconductors," in *Managing Temperature Effects in Nanoscale Adaptive Systems*. New York, NY: Springer New York, 2012, pp. 15–33.

[25] P. G. Neudeck, R. S. Okojie, and L.-Y. Chen, "High-temperature electronics-a role for wide bandgap semiconductors?" *Proceedings of the IEEE*, vol. 90, no. 6, pp. 1065–1076, 2002.

[26] C. Park, J. P. John, K. Klein, J. Teplik, J. Caravella, J. Whitfield, K. Papworth, and S. Cheng, "Reversal of temperature dependence of integrated circuits operating at very low voltages," in *Electron Devices Meeting, 1995. IEDM'95., International*. IEEE, 1995, pp. 71–74.

[27] F. Fallah, "Standby and Active Leakage Current Control and Minimization in CMOS VLSI Circuits," *IEICE Transactions on Electronics*, vol. E88-C, no. 4, pp. 509–519, Apr. 2005.

[28] J. Black, "Electromigration - a brief survey and some recent results," *IEEE Transactions on Electron Devices*, vol. 16, no. 4, pp. 338–347, Apr. 1969.

[29] I. Englander, *The architecture of computer hardware, system software, and networking: an information technology approach*, 4th ed. Hoboken, NJ: Wiley, 2009.

[30] "ITRS 2.0 Home Page." [Online]. Available: http://www.itrs2.net/

[31] J. Meena, S. Sze, U. Chand, and T.-Y. Tseng, "Overview of emerging nonvolatile memory technologies," *Nanoscale Research Letters*, vol. 9, 2014.

[32] Yole Developpement, "Emerging Non Volatile Memory Technology and Market Report," Jan. 2015. [Online]. Available: http://www.yole.fr/iso_album/illus_technicalchoices_nvm_jan2015.jpg

[33] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, *Flash Memories*. Boston, MA: Springer US, 1999.

[34] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," *Proceedings of the IEEE*, vol. 91, no. 4, pp. 489–502, Apr. 2003.

[35] H. Li, "Modeling of Threshold Voltage Distribution in NAND Flash Memory: A Monte Carlo Method," *IEEE Transactions on Electron Devices*, vol. 63, no. 9, pp. 3527–3532, Sep. 2016.

[36] Allen Kent and James G. Williams, *Encyclopedia of microcomputers - Volume 9*. New York and Basel: Marcel Dekker, INC., 1992.

[37] T. Heijmen, P. Roche, G. Gasiot, and K. Forbes, "A Comparative Study on the Soft-Error Rate of Flip-Flops from 90-nm Production Libraries." IEEE, 2006, pp. 204–211.

[38] A. S. Sedra and K. C. Smith, *Microelectronic circuits*, 5th ed., ser. The Oxford series in electrical and computer engineering. New York, NY: Oxford Univ. Press, 2004.

[39] G. Apostolidis, D. Balobas, and N. Konofaos, "Design and simulation of 6t SRAM cell architectures in 32nm technology," *Journal of Engineering Science and Technology Review*, vol. 9, no. 5, pp. 145–149, 2016.

[40] Georgios Tsiligiannis, Ioana Elena Vatajelu, Luigi Dilillo, Alberto Bosio, Serge Pravossoudovitch, Aida Todri-Sanial, Arnaud Virazel, Frederic Wrobel, and Frederic Saigne, "SRAM soft error rate evaluation under atmospheric neutron radiation and PVT variations," in *On-Line Testing Symposium (IOLTS), 2013 IEEE 19th International*. IEEE, 2013, pp. 145–150.

[41] E.I Vatajelu, G. Tsiligiannis, L. Dilillo, A. Bosio, P. Girard, and S. Pravossoudovitch, "On the correlation between Static Noise Margin and Soft Error Rate evaluated for a 40nm SRAM cell," *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*.

[42] N. Edri, S. Fraiman, A. Teman, and A. Fish, "Data retention voltage detection for minimizing the standby power of SRAM arrays." IEEE, Nov. 2012, pp. 1–5.

[43] Hulfang Qin, Yu Cao, D. Markovic, A. Vladimirescu, and J. Rabaey, "SRAM leakage suppression by minimizing standby supply voltage." IEEE Comput. Soc, 2004, pp. 55–60.

[44] S. Tavernier, *Experimental Techniques in Nuclear and Particle Physics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[45] J.-L. Autran, S. Semikh, D. Munteanu, S. Serre, G. Gasiot, and P. Roche, "Soft-Error Rate of Advanced SRAM Memories: Modeling and Monte Carlo Simulation," in *Numerical Simulation - From Theory to Industry*, M. Andriychuk, Ed. InTech, Sep. 2012.

[46] Microsemi, "Neutron-induced single event upsets (SEU) FAQ," Tech. Rep., Aug. 2011.

[47] R. Antoni and L. Bourgois, "Quantities and Fundamental Units of External Dosimetry," in *Applied Physics of External Radiation Exposure*. Cham: Springer International Publishing, 2017, pp. 1–42.

[48] O. U. Press, "Icru report 85," *Journal of the International Commission on Radiation Units and Measurements*, vol. 11, no. 1, 2011.

[49] E. Petersen, *Single Event Effects in Aerospace*. John Wiley & Sons, Nov. 2011.

[50] D. J. DiMaria and E. Cartier, "Mechanism for stressâĂŘinduced leakage currents in thin silicon dioxide films," *Journal of Applied Physics*, vol. 78, no. 6, pp. 3883–3894, Sep. 1995.

[51] E. Rosenbaum, "Mechanism of Stress-Induced Leakage Current in MOS Capacitors," *IEEE TRANSACTIONS ON ELECTRON DEVICES*, vol. 44, no. 2, p. 7, 1997.

[52] D. Sheldon and M. Freie, "Disturb Testing in Flash Memories," National Aeronautics and Space Administration, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, Tech. Rep. JPL Publication 08-7 3/08.

[53] J. Heidecker, "NAND Flash Screening and Qualification Guideline for Space Application," National Aeronautics and Space Administration, Jet Propulsion Laboratory, California Institute of Technology, Tech. Rep. JPL Publication 12-1 2/12.

[54] D. Bertozzi, S. D. Carlo, S. Galfano, M. Indaco, P. Olivo, P. Prinetto, and C. Zambelli, "Performance and Reliability Analysis of Cross-Layer Optimizations of NAND Flash Controllers," *ACM Transactions on Embedded Computing Systems*, vol. 14, no. 1, pp. 1–24, Jan. 2015.

[55] S. D. Carlo, S. Galfano, M. Indaco, P. Prinetto, D. Bertozzi, P. Olivo, and C. Zambelli, "FLARES: An Aging Aware Algorithm to Autonomously Adapt the Error Correction Capability in NAND Flash Memories," *ACM Transactions on Architecture and Code Optimization*, vol. 11, no. 3, pp. 1–25, Jul. 2014.

[56] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling." IEEE Conference Publications, 2013, pp. 1285–1290.

[57] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, and O. Mutlu, "Data retention in MLC NAND flash memory: Characterization, optimization, and recovery." IEEE, Feb. 2015, pp. 551–563.

[58] Y. Cai, G. Yalcin, O. Mutlu, E. F. Haratsch, A. Cristal, O. S. Unsal, and K. Mai, "Flash correct-and-refresh: Retention-aware error management for increased flash memory lifetime." IEEE, Sep. 2012, pp. 94–101.

[59] P. E. Dodd, M. R. Shaneyfelt, J. R. Schwank, and J. A. Felix, "Current and Future Challenges in Radiation Effects on CMOS Electronics," *IEEE Transactions on Nuclear Science*, vol. 57, no. 4, pp. 1747–1763, Aug. 2010.

[60] J.-L. Autran, S. Semikh, D. Munteanu, S. Serre, G. Gasiot, and P. Roche, "Soft-Error Rate of Advanced SRAM Memories: Modeling and Monte Carlo Simulation," in *Numerical Simulation - From Theory to Industry*, M. Andriychuk, Ed. InTech, Sep. 2012.

[61] L.H. Mutuel, "Single Event Effects Mitigation Techniques Report," Federal Aviation Administration, William J. Hughes Technical Center, Aviation Research Division, Atlantic City International Airport, Final report DOT/FAA/TC-15/62, Feb. 2016.

[62] C. Slayman, "Soft error trends and mitigation techniques in memory devices," in *2011 Proceedings - Annual Reliability and Maintainability Symposium*. Lake Buena Vista, FL, USA: IEEE, Jan. 2011, pp. 1–5.

[63] F.-X. Yu, J.-R. Liu, Z.-L. Huang, H. Luo, and Z.-M. Lu, "Overview of radiation hardening techniques for ic design," *Information Technology Journal*, vol. 9, no. 6, pp. 1068–1080, 2010.

[64] M. Haghi and J. Draper, "The 90 nm Double-DICE storage element to reduce Single-Event upsets," in *Circuits and Systems, 2009. MWSCAS'09. 52nd IEEE International Midwest Symposium on*. IEEE, 2009, pp. 463–466.

[65] L.-P. Chang, "Hybrid solid-state disks: Combining heterogeneous nand flash in large ssds," in *2008 Asia and South Pacific Design Automation Conference*, March 2008, pp. 428–433.

[66] Y. Chen, "Flash Memory Reliability NEPP 2008 Task Final Report," National Aeronautics and Space Administration, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, Tech. Rep. JPL Publication 09-9 3/09, 2008.

[67] JEDEC SOLID STATE TECHNOLOGY ASSOCIATION, "JEDEC PUBLICATION: Failure Mechanisms and Models for Semiconductor Devices," Tech. Rep. JEP122F (Revision of JEP122E, March 2009), Nov. 2010.

[68] ——, "JEDEC STANDARD: Solid-State Drive (SSD) Requirements and Endurance Test Method," Tech. Rep. JESD218B.01, Jun. 2016.

[69] V. Mohan, S. Sankar, S. Gurumurthi, and W. Redmond, "reFresh SSDs: Enabling high endurance, low cost flash in datacenters," *Univ. of Virginia, Tech. Rep. CS-2012-05*, 2012.

[70] Micron Technology, Inc., "Uprating Semiconductors for High-Temperature Applications," Technical note TN-00-18, 2004.

[71] P. Ellerman, "Calculating Reliability using FIT & MTTF: Arrhenius HTOL Model," Microsemi corp., MicroNote 1002, 2012.

[72] K. Lee, M. Kang, S. Seo, D. H. Li, J. Kim, and H. Shin, "Analysis of Failure Mechanisms and Extraction of Activation Energies Ea in 21-nm nand Flash Cells," *IEEE Electron Device Letters*, vol. 34, no. 1, pp. 48–50, Jan. 2013.

[73] K. Lee, M. Kang, S. Seo, D. Kang, S. Kim, D. H. Li, and H. Shin, "Activation Energies Ea of Failure Mechanisms in Advanced NAND Flash Cells for Different Generations and Cycling," *IEEE Transactions on Electron Devices*, vol. 60, no. 3, pp. 1099–1107, Mar. 2013.

[74] N. Sundby and D. Taylor, "Beyond Capacity: Storage Architecture Choices for the Modern Datacenter," White paper - Toshiba, Feb. 2014.

[75] Kingston Technology Company, "The Difference Between Enterprise & Client SSD | Kingston." [Online]. Available: https://www.kingston.com/us/ssd/enterprise/best_practices/enterprise_versus_client_ssd

[76] J. Meza, Q. Wu, S. Kumar, and O. Mutlu, "A Large-Scale Study of Flash Memory Failures in the Field," in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems - SIGMETRICS '15*. Portland, Oregon, USA: ACM Press, 2015, pp. 177–190.

[77] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, "Errors in flash-memory-based solid-state drives: Analysis, mitigation, and recovery," *CoRR*, vol. abs/1711.11427, 2017. [Online]. Available: http://arxiv.org/abs/1711.11427

[78] Reliability HotWire, Reliability e-Magazine, Issue 116, "Chi-Squared Distribution and Reliability Demonstration Test Design," Oct. 2010. [Online]. Available: https://www.weibull.com/hotwire/issue116/relbasics116.htm

[79] Paul Ellerman, Microsemi corp, "Calculating Chi-squared for Reliability Equations," Tech. Rep. MicroNote 1003, Sep. 2012.

[80] N. Kr.Shukla, "Analysis of the Effect of Temperature Variations on Sub-threshold Leakage Current in P3 and P4 SRAM Cells at Deep Sub-micron CMOS Technology," *International Journal of Computer Applications*, vol. 35, no. 5, pp. 8–13, Dec. 2011.

[81] Hulfang Qin, Yu Cao, D. Markovic, A. Vladimirescu, and J. Rabaey, "SRAM leakage suppression by minimizing standby supply voltage," in *SCS 2003. International Symposium on Signals, Circuits and Systems. Proceedings (Cat. No.03EX720)*. San Jose, CA, USA: IEEE Comput. Soc, 2004, pp. 55–60.

[82] N. Edri, S. Fraiman, A. Teman, and A. Fish, "Data retention voltage detection for minimizing the standby power of SRAM arrays," in *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*. Eilat, Israel: IEEE, Nov. 2012, pp. 1–5.

[83] G. Tsiligiannis, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Todri-Sanial, A. Virazel, C. Frost, F. Wrobel, and F. Saigné, "Temperature Impact on the Neutron SER of a Commercial 90nm SRAM," in *NSREC: Nuclear and Space Radiation Effects Conference.* San Francisco, Ca, United States: IEEE, Jul. 2013, pp. 1–4.

[84] G. Tsiligiannis, E. I. Vatajelu, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Todri, A. Virazel, F. Wrobel, and F. Saigne, "Sram soft error rate evaluation under atmospheric neutron radiation and pvt variations," in *2013 IEEE 19th International On-Line Testing Symposium (IOLTS)*, July 2013, pp. 145–150.

[85] M. Bagatin, S. Gerardin, A. Paccagnella, C. Andreani, G. Gorini, and C. Frost, "Temperature dependence of neutron-induced soft errors in SRAMs," *Microelectronics Reliability*, vol. 52, no. 1, pp. 289–293, Jan. 2012.

[86] D. Truyen, J. Boch, B. Sagnes, N. Renaud, E. Leduc, S. Arnal, and F. Saigne, "Temperature Effect on Heavy-Ion Induced Parasitic Current on SRAM by Device Simulation: Effect on SEU Sensitivity," *IEEE Transactions on Nuclear Science*, vol. 54, no. 4, pp. 1025–1029, Aug. 2007.

[87] R. Naseer, Y. Boulghassoul, J. Draper, S. DasGupta, and A. Witulski, "Critical Charge Characterization for Soft Error Rate Modeling in 90nm SRAM," in *2007 IEEE International Symposium on Circuits and Systems.* New Orleans, LA, USA: IEEE, May 2007, pp. 1879–1882.

[88] T. Liu, C. Geng, Z. Zhang, F. Zhao, S. Gu, T. Tong, K. Xi, G. Liu, Z. Han, M. Hou, and J. Liu, "Impact of temperature on single event upset measurement by heavy ions in SRAM devices," *Journal of Semiconductors*, vol. 35, no. 8, Aug. 2014.

[89] G. R. Srinivasan, P. C. Murley, and H. K. Tang, "Accurate, predictive modeling of soft error rate due to cosmic rays and chip alpha radiation," in *Proceedings of 1994 IEEE International Reliability Physics Symposium*, April 1994, pp. 12–16.

[90] Balkaran Gill, M. Nicolaidis, F. Wolff, C. Papachristou, and S. Garverick, "An Efficient BICS Design for SEUs Detection and Correction in Semiconductor Memories," in *Design, Automation and Test in Europe.* Munich, Germany: IEEE, 2005, pp. 592–597.

[91] F. Wrobel, L. Dilillo, A. D. Touboul, V. Pouget, and F. Saigne, "Determining Realistic Parameters for the Double Exponential Law that Models Transient Current Pulses," *IEEE Transactions on Nuclear Science*, vol. 61, no. 4, pp. 1813–1818, Aug. 2014.

[92] F. Wrobel, L. Dilillo, A. D. Touboul, and F. Saigne, "Comparison of the transient current shapes obtained with the diffusion model and the double exponential law - Impact on the SER," in *2013 14th European Conference on Radiation and Its Effects on Components and Systems (RADECS).* Oxford, United Kingdom: IEEE, Sep. 2013, pp. 1–4.

[93] G. Torrens, "FPGA SRAM Soft Error Radiation Hardening," in *Field - Programmable Gate Array*, G. Dekoulis, Ed. InTech, May 2017.

[94] A. Maru, H. Shindou, T. Ebihara, A. Makihara, T. Hirao, and S. Kuboyama, "DICE-Based Flip-Flop With SET Pulse Discriminator on a 90 nm Bulk CMOS Process," *IEEE Transactions on Nuclear Science*, Dec. 2010.

[95] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE Journal of solid-state circuits*, vol. 22, no. 5, pp. 748–754, 1987.

[96] "Predictive Technology Model (PTM)." [Online]. Available: http://ptm.asu.edu/

[97] Y. Cao, "Predictive Technology Model of Conventional CMOS Devices," in *Predictive Technology Model for Robust Nanoelectronic Design*. Boston, MA: Springer US, 2011, pp. 7–23.