Technische Universität München

Lehrstuhl für Entwurfsautomatisierung

Prediction, Mitigation, and Emulation of Reliability Risks in System-on-Chips for Advanced Technology Nodes

Alexandra Listl, M.Sc.

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades einer

Doktorin der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitz:		Prof. Dr.-Ing. Georg Sigl
Prüfer der Dissertation:	1.	Prof. Dr.-Ing. Ulf Schlichtmann
	2.	apl. Prof. Dr.-Ing. Walter Stechele

Die Dissertation wurde am 03.03.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 15.10.2021 angenommen.

Abstract

Relentless technology scaling has led to higher power dissipation, rising on-chip temperatures, and an increasing impact of transistor wear-out mechanisms which put product reliability at high risk. Reliability assessment during the design phase of the product is more important than ever to reduce costs and guarantee that reliability specifications are met. The scope of this thesis is, therefore, to predict, mitigate and emulate arising reliability threats caused by power consumption, temperature, and aging in two major building blocks of modern System-on-Chips (SoCs): the on-chip SRAM and the Central Processing Unit (CPU). Therefore, the impact of Bias Temperature Instability (BTI) as a dominant aging mechanism in 32nm and 40nm CMOS technologies is regarded.

In the first part of this work, the effect of BTI is analyzed with a novel reliability tool for SRAM Design-for-Reliability. The developed tool (AppAwareAge) incorporates a new workload-aware aging analysis to predict the aging-induced degradation of on-chip SRAMs during the design phase based on the workload of embedded applications executed on an industrial Micro-Controller Unit (MCU). An application-aware end-of-life analysis of the SRAM can predict the expected lifetime for the given workload and operating conditions.

Furthermore, this work introduces, as a first aging mitigation technique the Mitigation of AGIng Circuitry (MAGIC), a low-cost circuitry to effectively mitigate aging in Sense Amplifiers (SAs) by wear-leveling. MAGIC modifies the mapping of SRAM banks to physical addresses and distributes the stress onto the complete SRAM array to avoid exacerbated aging in highly-used addresses. Deploying the proposed reliability tool, an extensive study of an industrially used SRAM design compares the aging behavior of the read-path with and without the proposed mitigation technique for various workloads, temperatures, and supply voltages. Since the developed tool is capable of analyzing SRAM architectures of arbitrary size and granularity, the second aging mitigation technique utilizes the tool as an SRAM design exploration framework (SDE) that generates and characterizes memories of different array granularity (i.e. number of banks/rows/words) with detailed simulations. Since the array granularity has a notable impact on the aging rates of the memory, aging can be effectively mitigated by exploring the most reliable configuration for a given set of applications early in the design phase.

The second part of this work focuses on the reliability assessment during the design phase of CPUs and therefore proposes a real-time power, temperature, and aging monitor system (eTAPMon) for FPGA prototypes of Multi-Processor System-on-Chips (MPSoCs). The monitoring approach can be used to emulate the behavior of ASIC monitors on an FPGA prototyping platform in order to develop efficient runtime management and resource allocation strategies. The monitor system was implemented on an FPGA board and evaluated for a selected operating scenario for nominal process corners, where it provides useful insights into the power, temperature, and aging behavior of the system.

Zusammenfassung

Die unermüdliche Skalierung der CMOS-Technologie führt zu höheren Verlustleistungen, steigenden On-Chip-Temperaturen und einem zunehmenden Einfluss von Transistorverschleißmechanismen, welche ein wachsendes Risiko für die Produktzuverlässigkeit darstellen. Eine Analyse der Zuverlässigkeit während der Entwurfsphase des Produkts ist daher wichtiger denn je um Kosten zu senken, und die Einhaltung von Zuverlässigkeitsspezifikationen zu gewährleisten. Ziel dieser Arbeit ist es daher, auftretende Zuverlässigkeitsrisiken aufgrund von Leistungsverbrauch, Temperatur und Alterung in zwei Hauptbausteinen moderner System-on-Chips (SoCs) vorherzusagen, zu mindern und zu emulieren: dem On-Chip-SRAM und der Central Processing Unit (CPU). Dabei wird der Einfluss von Bias Temperature Instability (BTI) als einer der dominanten Alterungsmechanismen in 32nm und 40nm CMOS-Technologien betrachtet.

Im ersten Teil dieser Arbeit wird die Wirkung von BTI mit einem neuen Zuverlässigkeitstool für SRAM Design-for-Reliability analysiert. Das entwickelte Tool (AppAwareAge) enthält eine neue arbeitslastberücksichtigende Alterungsanalyse, um die alterungsbedingte Degradation von On-Chip-SRAMs während der Entwurfsphase basierend auf der Arbeitlast eingebetteter Anwendungen vorherzusagen, welche auf einer industriellen Micro-Controller Unit (MCU) ausgeführt werden. Eine End-of-Life-Analyse des SRAM kann die erwartete Lebensdauer für gegebene Arbeitslast und Betriebsbedingungen vorhersagen. Zudem wird in dieser Arbeit als erste Methode zur Abschwächung der Alterung Mitigation of AGIng Circuitry (MAGIC) vorgestellt, eine kostengünstige Schaltung zur wirksamen Minderung der Alterung in Sense Amplifiern (SAs) durch Wear-leveling. MAGIC modifiziert die Zuordnung von SRAM-Bänken zu physikalischen Adressen und verteilt die Arbeitslast auf das gesamte SRAM-Array, um eine verstärkte Alterung bei häufig verwendeten Adressen zu vermeiden. Unter Verwendung des Zuverlässigkeitstools vergleicht eine Studie eines industriell verwendeten SRAM-Designs das Alterungsverhalten des Lesepfads mit und ohne die Technik zu Minderung der Alterung. Da das entwickelte Tool in der Lage ist SRAM-Architekturen beliebiger Größe und Granularität zu analysieren, verwendet eine zweite Technik zur Abschwächung der Alterung das Tool als SRAM Design Exploration Framework (SDE), welche Speicher verschiedener Array-Granularität (Anzahl der Bänke/ Zeilen/ Wörter) generiert und charakterisiert. Da die Array-Granularität einen starken Einfluss auf die Alterungsraten des Speichers hat, kann die Alterung wirksam gemindert werden, indem die zuverlässigste Konfiguration für bestimmte Anwendungen zu Beginn der Entwurfsphase untersucht wird.

Der zweite Teil dieser Arbeit konzentriert sich auf die Zuverlässigkeitsanalyse während der Entwurfsphase von CPUs und stellt dafür ein Echtzeit-Monitoring-System für Leistung, Temperatur und Alterung (eTAPMon) für FPGA-Prototypen von Multi-Prozessor-System-on-Chips (MPSoCs) vor. Der Monitoring-Ansatz kann verwendet werden, um das Verhalten von ASIC-Monitoren auf einer FPGA-Prototyping-Plattform zu emulieren und effiziente Strategien für das Runtime-Management und die Zuweisung von Prozessorressourcen zu entwickeln. Das Monitoring-System wurde auf einem FPGA-Board implementiert und für ein ausgewähltes Betriebsszenario bewertet.

Publications by the Author

Conferences

A. Listl, D. Mueller-Gritschneder, U. Schlichtmann and S. R. Nassif, “SRAM Design Exploration with Integrated Application-Aware Aging Analysis,” *2019 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1249-1252, 2019.

A. Listl, D. Mueller-Gritschneder and U. Schlichtmann, “MAGIC: A Wear-leveling Circuitry to Mitigate Aging Effects in Sense Amplifiers of SRAMs,” *2019 17th IEEE International New Circuits and Systems Conference (NEWCAS)*, pp. 1-4, 2019.

A. Listl, D. Mueller-Gritschneder, F. Kluge and U. Schlichtmann, “Emulation of an ASIC Power, Temperature and Aging Monitor System for FPGA Prototyping,” *2018 IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS)*, pp. 220-225, 2018.

L. Zhang, A. Listl, B. Li, U. Schlichtmann, “Effizienter Verzögerungstest zur Optimierung der Taktfrequenz einer Schaltung durch nach der Fertigung konfigurierbare Puffer,” *edaworkshop*, 2016.

N. P. Aryan, A. Listl, L. Heiss, C. Yilmaz, G. Georgakos and D. Schmitt-Landsiedel, “From an analytic NBTI device model to reliability assessment of complex digital circuits,” *2014 IEEE 20th International On-Line Testing Symposium (IOLTS)*, pp. 19-24, 2014.

Journals

M. Glaß, H. Aliee, L. Chen, M. Ebrahimi, F. Khosravi, V.B. Kleeberger, A. Listl, D. Müller-Gritschneder, F. Oboril, U. Schlichtmann, M.B. Tahoori, J. Teich, N. Wehn and C. Weis, “Application-aware cross-layer reliability analysis and optimization,” *it - Information Technology*, vol.57, nr.3, pp. 159-169, 2015.

Patents

A. Listl, D. Mueller-Gritschneder, *Verfahren und Vorrichtung zum Betreiben einer Speicheranordnung*. PCT Patent, Veröffentlichungsnr.: WO 2020/104091, eingereicht 2019.

A. Listl, D. Mueller-Gritschneder, *Verfahren und Vorrichtung zum Betreiben einer Speicheranordnung*. Deutsches Patent- und Markenamt, Veröffentlichungsnr.: 10 2018 128 980, eingereicht 2018.

Contents

1 Motivation **1**
- 1.1 Introduction . . . 1
- 1.2 Aging Prediction and Mitigation in On-Chip SRAMs . . . 3
- 1.3 Emulation of ASIC Hardware Monitors on FPGA Prototypes . . . 4
- 1.4 Contributions . . . 5

2 State of the Art **9**
- 2.1 Aging Analysis and Mitigation Techniques for SRAMs . . . 9
  - 2.1.1 Aging Analysis of SRAMs . . . 9
  - 2.1.2 Aging Mitigation in SRAM Core Cells . . . 10
  - 2.1.3 Aging Mitigation in the Sense Amplifiers . . . 11
- 2.2 Power, Temperature and Aging Emulation in Multi-Core Processors 12
  - 2.2.1 Power Emulation . . . 12
  - 2.2.2 Temperature Emulation . . . 13
  - 2.2.3 Aging Emulation . . . 13
  - 2.2.4 Enhancement of the State of the Art . . . 14

3 Reliability and Degradation **15**
- 3.1 Failure Rate Curve . . . 16
- 3.2 Electron Device Physics of Failure . . . 19
  - 3.2.1 Negative Bias Temperature Instability . . . 19
  - 3.2.2 Positive Bias Temperature Instability . . . 20
  - 3.2.3 Analytical Bias Temperature Instability (BTI) Models . . . 21
  - 3.2.4 Other Degradation Mechanisms . . . 23
- 3.3 Summary . . . 24

4 Application-Aware Aging Analysis and Mitigation for on-Chip SRAM Design-for-Reliability **27**
- 4.1 SRAM Circuit Design and Operation . . . 29
  - 4.1.1 SRAM Architecture . . . 31
  - 4.1.2 Circuit Design and Operation of the 6T-SRAM Core Cell . . . 34

4.1.3 Circuit Design and Operation of the Voltage-Latch Sense Amplifier . . . 35
4.2 Important SRAM Performance Parameters and Figures of Merit . . 38
4.2.1 Static Noise Margins . . . 38
4.2.2 Dynamic Noise Margins . . . 40
4.3 BTI Aging of Sense Amplifiers (SAs) and Cells . . . 44
4.3.1 The BTI Model . . . 44
4.3.2 BTI Workload Parameters . . . 44
4.3.3 BTI Aging of the Sense Amplifiers (SAs) . . . 46
4.3.4 BTI Aging of 6T SRAM Cells . . . 48
4.4 Application-aware Analysis of BTI Aging (AppAwareAge) . . . 49
4.4.1 Stress Profiling by High-level Simulation . . . 50
4.4.2 Aging Analysis by low-level Transistor Simulation . . . 52
4.4.3 Speeding-up the Analysis . . . 54
4.5 Aging Mitigation in SRAMs . . . 55
4.5.1 Mitigation of AGIng Circuitry (MAGIC) . . . 55
4.5.2 Aging-Aware SRAM Design Exploration (SDE) . . . 59
4.6 Summary . . . 62

5 Runtime Power, Temperature and Aging Monitoring on FPGA Prototypes 63
5.1 Invasive Computing . . . 65
5.2 FPGA Prototyping in Multi-Core Processors . . . 67
5.3 ASIC Monitoring System . . . 68
5.3.1 Emulation of the Power Monitor . . . 69
5.3.2 Emulation of the Temperature Monitor . . . 71
5.3.3 Emulation of the Aging Monitor . . . 73
5.4 Summary . . . 77

6 Experimental Results 79
6.1 Application-Aware Aging Analysis (AppAwareAge) and Mitigation of AGIng Circuitry (MAGIC) . . . 79
6.1.1 Experimental Setup . . . 79
6.1.2 Memory Utilization and Application Stress Profiles . . . 81
6.1.3 Analysis of the individual Contributions of Cell and SA Aging on the Read-Path Degradation using AppAwareAge 82
6.1.4 Effectiveness of MAGIC . . . 90
6.1.5 Lifetime Prediction without and with MAGIC . . . 98
6.1.6 Area Overhead . . . 99

6.2 SRAM Design Exploration (SDE) 100
6.2.1 Experimental Setup 100
6.2.2 Stress Profiles of all possible Memory Configurations 100
6.2.3 Read-Path Degradation of all possible Memory Configurations 105
6.2.4 Area Overhead 107
6.3 Runtime Power, Temperature and Aging Monitoring on FPGA Prototypes (eTAPMon) 107
6.3.1 Experimental Setup 107
6.3.2 Emulation Results from the Implementation of eTAPMon on an FPGA Board 107
6.3.3 Hardware Overhead 110
6.4 Summary 112

7 Conclusion and Future Work **113**

List of Figures **116**

List of Tables **119**

Bibliography **123**

1 Motivation

1.1 Introduction

Integrated Circuits (ICs) have unleashed a radical change on humankind and brought forth a great revolution by powering the computer and digital age. The key driver of this rapid progress has been Moore's Law, which essentially states that the number of components per chip approximately doubles every two years leading to higher integration densities and more functionality within the same die size [1]. In combination with rising transistor speeds, the processing power of ICs has continuously grown to sustain increasingly more complex chips. With the ability of the IC industry to follow Moore's law, progress was rapid and nourished major advances, from smartphones and the Internet of Things (IoT) to automotive electronics and artificial intelligence. ICs are connecting the world, automating everyday life and increasing the quantity and ease of access to information for everyone.

The continuous increase in the integration density proposed by Moore's Law was enabled by a dimensional scaling known as Dennard Scaling [2]. The concept implies that reduced physical parameters of transistors allows them to be operated at lower voltage while preserving constant power density. This type of scaling is generally known as constant-field scaling. Because the device dimensions and the supply voltage are scaled uniformly, the electrical field over the channel stays constant while the power consumption per area effectively reduces and operating speeds increase [3]. Unfortunately, in the deep sub-micron regime this constant-field scaling has reached an end, since the supply voltage cannot be further scaled without introducing a significant increase of leakage power consumption.

Consequently, electrical fields increase and the power consumption of transistors is not scaling as quickly as the transistor dimensions leading to an increase in both power density and total power consumption for a fixed-size chip [4]. Hence, it is not surprising that power consumption is now becoming the limiting factor

since most computer systems must function within a given power envelope, the Thermal Design Power (TDP). As a consequence of the power problem, a temperature problem arises, as higher power density and larger leakage currents ultimately heat the chip intolerably. Physical limits imposed device packaging and cooling technology introduce hard constraints on the maximum amount of heat that can be removed from the chip. In combination with limitations imposed by battery technology, the maximum power that can be supplied to a chip is severely restricted. While the computational capabilities of chips are still increasing according to Moore's Law, the power wall created by the breakdown of Dennard Scaling will soon prevent us from powering all transistors at the full performance level simultaneously, resulting in the so-called "dark silicon problem", leaving a large fraction of the chip powered off (dark) or underclocked (dim) [5].

The current trend of increasing electrical fields, rising temperatures as well as the deployment of new materials has given birth to new undesirable effects of transistor wear-out. These time-dependent variations cause a degradation in the electrical properties of a transistor over its lifetime. Several effects can lead to device aging such as Negative and Positive Bias Temperature Instability (NBTI/PBTI), Hot Carrier Injection (HCI), Electromigration (EM) and Time-Dependent Dielectric Breakdown (TDDB). BTI has been identified as one of the most important degradation mechanisms in modern CMOS technologies. BTI gradually increases the threshold voltage (V_{th}) of a transistor and degrades the drain current, thus leading to poorer drive currents, increasing logic gate delays and lower noise margins which ultimately threaten the reliable functionality of a circuit over time.

Another major concern that arises through transistor scaling below the sub-nm regime are process variations. They originate from imperfections in the manufacturing process since the ability to control important transistor parameters rapidly diminishes for shrinking feature sizes. Variability in the physical parameters of a transistor causes unpredictability of the electrical parameters, which affect the performance and timing characteristics of nanometer circuits and considerably impair yield. Sources for device variability include, e.g., variations in the transistor's dimensions, random dopant fluctuations as well as line edge roughness. Inevitably, process variations aggravate reliability concerns, since they appear in addition to transistor wear-out.

The recent scaling trends have precipitated an inflection point in which optimizing on-chip systems for reliability is at least as important as optimizing them for

performance and yield [6]. In modern IC designs, typically a large fraction of the die area is occupied by memory and processor logic. With both of these major building blocks running into considerable reliability problems at advanced technology nodes, the scope of this thesis is the reliability assessment of vastly used on-chip Static Random Access Memories (SRAMs) as well as embedded processors of Multi-Processor System-on-Chips (MPSoCs) and to develop methods to predict, mitigate and emulate arising reliability threats caused by power, temperature and aging.

1.2 Aging Prediction and Mitigation in On-Chip SRAMs

The discussed advances in technology have enabled the integration of all components and functions of electronic systems into a single silicon chip. On-chip memories are an integral part of these so called System-on-Chips (SoCs) and hence SoC performance depends heavily on the storage capacity and access speed of their embedded memories. With today's increasing memory demands, memory blocks already occupy more than 80% of the transistor count and up to 90% of the chip's area [7]. In virtue of future memory-hungry applications, like high-end and mobile computing, augmented reality and artificial intelligence, this share is expected to increase drastically. Due to their speed advantage and compatibility with the CMOS process technology, SRAMs are currently the most dominant on-chip memory type in SoCs. At the same time, however, SRAMs are especially vulnerable to process, temperature, voltage and aging (PVTA) variability, since they are aggressively scaled and exhibit the highest integration densities.

The traditional approach to compensate variability on circuit level is to introduce guardbands which add additional safety margins either with respect to timing, by adding sufficient time to the maximum delay of the critical path or with respect to the supply voltage, by adding sufficient voltage safety margins to the nominal minimum voltage [8]. The first guarbanding technique thereby results in a performance loss in terms of frequency, since effectively the maximum achievable operating frequency of the circuit is lowered. The latter technique however creates a power overhead and therefore exacerbates the problems related to dark silicon [6]. Usually, these design margins are based on the worst-case scenario in state-of-the-art designs. This worst-case approach prepares for the worst-case combination of PVTA conditions and workloads and guarantees

proper operation in extremely sub-optimal scenarios. The term workload defines the portion of lifetime an individual transistors experiences aging stress and hence the extent to which it is affected by aging.

Such conditions however rarely occur and therefore lead to excessively large guardbands for standard operating conditions at the expense of more area, power and lower speed. This is especially true for aging guardbands, since aging is heavily dependent on the workload [9, 10] and worst-case workload scenarios rarely match the workload that is induced by real applications. Since PVTA guardbands need to be stacked on top of each other, an augmentation of the guardbands, e.g. to compensate for an increasing aging variability in scaled technology nodes, needs to be kept at a minimum. An accurate prediction of aging guardbands for a given SRAM architecture and device technology is thus essential to avoid an overestimation of the necessary safety margin and hence an inefficient circuit design. Insufficient aging guardbands on the other hand ultimately endanger the reliability of a circuit. A precise prediction of the actual degradation through aging analysis methods for on-chip SRAMs which accurately capture the appearing workload, temperature and supply voltage are therefore essential to estimate the correct amount of necessary margins to compensate aging variability.

As a more cost-effective alternative to safety margins, mitigation schemes can be applied to counteract aging. Such mitigation schemes can avoid unbalanced aging and hence significantly improve aging-induced transistor wear-out. Although many mitigation schemes can effectively curb wasteful guardbands, they usually come with a considerable cost in terms of power and area.

1.3 Emulation of ASIC Hardware Monitors on FPGA Prototypes

When processors were hitting the power wall and approaching the limits of cooling technologies, the on-chip frequency growth halted and micro-architectural techniques alone were not enough to maintain the increasing performance demand according to Moore's Law. Hence, a direct consequence of the breakdown of Dennard Scaling was the transition to Multi-Processor Systems-on-Chips (MPSoCs). MPSoCs exploit computational parallelism instead of frequency-scaling and are so far able to compensate the ever growing demand for performance.

However, multi-core alone is not the final solution to dark silicon [11]. ITRS projections have predicted that designers will face up to 90% of dark silicon in the near future when high operating frequencies are applied [12]. Keeping up with the expected performance and efficiency demands requires further technology scaling to be able to support future computationally intensive applications like deep machine learning, virtual reality and big data which ultimately lead to a further rise in power densities. Hence, the dark silicon issue has only been postponed through the introduction of MPSoCs. It is projected that multi-core scaling is soon just as limited by the dark silicon problem as single-core scaling.

Architectural and runtime management techniques on heterogeneous many- and multi-core systems can mitigate the dark silicon problem and improve energy efficiency. In combination with resource-aware computing concepts, they offer the chance to dynamically control and distribute resources among different applications running on a single chip in order to satisfy high resource utilization and energy efficiency to counter the challenges of dark silicon [13].

Here, monitoring data provides crucial information about the current hardware health and can hence be applied during system runtime to adjust resource utilization and performance to improve system lifetime and reliability [14]. To evaluate and optimize runtime management and resource allocation strategies during the design phase, FPGA prototyping, which is already a well-established method for functional verification and early software development, can be utilized before the implementation of the ASIC. Through the placement of hardware monitors on FPGA prototypes, the necessary data needed to develop efficient load distributions, operating strategies and control targets for an efficient resource-aware computing and runtime management can be acquired even if the ASIC does not exist yet.

1.4 Contributions

This thesis investigates methods to predict, mitigate and emulate arising reliability threats caused by power consumption, temperature and aging in two major building blocks of modern SoCs: the on-chip SRAM and the Central Processing Unit (CPU).

For the design phase of on-chip SRAMs, this work proposes a novel reliability tool (AppAwareAge) for SRAM Design-for-Reliability incorporating a new

workload-aware aging analysis for on-chip SRAMs. The tool investigates BTI and its recovery effect. It incorporates the workload of embedded applications executed on an industrial Micro-Controller Unit (MCU) while considering aging in the complete read-path and its control signals. According to this workload, the performance degradation in the memory and its end-of-life can be accurately predicted. Thus, the tool builds a bridge between high-level and low-level simulation methods. Applying the proposed reliability tool, the contributions of Sense Amplifiers (SAs) and SRAM cell aging to the degradation of the read-path of the SRAM for various workloads, temperatures and supply voltages is analyzed for an industrially used SRAM design. It is shown that realistic workloads are a vital factor for an accurate aging prediction and hence, SA aging was often overestimated in previous work due to the lack of accurate workload assumptions. Furthermore, depending on the sizing and the stress level, both SRAM cell and SA aging have a significant contribution to the overall degradation of the read-path while aging in the SA's control signals non-intuitively leads to minor performance improvements.

Furthermore, the Mitigation of AGIng Circuitry (MAGIC), a low-cost circuitry to effectively mitigate aging in SAs by wear-leveling is presented. MAGIC modifies the mapping of SRAM banks to physical addresses in order to distribute the memory accesses evenly over the complete SRAM array. The AppAwareAge tool is deployed to compare the aging behavior of the read-path with and without the proposed mitigation technique and demonstrate its effectiveness for various workloads, temperatures and supply voltages. The proposed mitigation scheme MAGIC can mitigate the degradation in the read-path up to 26% for three years of aging while introducing minimal area/performance overhead. An application-aware end-of-life analysis of the SRAM shows that this translates into 3x longer lifetime.

Moreover, the proposed tool can be used as an SRAM design exploration framework (SDE) that generates and characterizes memories of different array granularity (e.g. number of banks/rows/words) with detailed simulations to find the most reliable configuration in terms of aging for the intended set of applications. The presented results show that the array granularity has a significant impact on the aging behavior of the memory. SDE can improve SA degradation by up to 32% for three years of aging while showing a low area penalty. Hence, this tool can be a helpful means during the design phase of safety critical systems to predict the memory lifetime and mitigate aging by selecting the most reliable design for the intended set of applications.

For reliability assessment during the design phase of CPUs, another contribution of this work is a real-time power, temperature and aging monitor system (eTAPMon) for FPGA prototypes of MPSoCs. The monitor system is able to predict reliability threats and can supply emulated data characterized from the target ASIC design to the runtime power management to develop efficient power management and resource allocation strategies. For this purpose, a modeling approach was developed that can be used to emulate the behavior of ASIC monitors on an FPGA prototyping platform. The emulation approach models the behavior of ASIC power monitors based on an instruction-level energy model, the behavior of temperature monitors based on a linear regression model obtained from thermal offline simulations and the behavior of aging monitors based on a critical path model to compute the decreasing timing margin due to aging. An accelerated aging emulation is possible to predict aged ASIC behavior. Hence, this FPGA emulation enables the early evaluation of runtime management and resource allocation strategies. The monitor system was implemented on an FPGA board and evaluated for a selected operating scenario for nominal process corners, where it provides useful insights into the power, temperature and aging behavior of the system.

The rest of this thesis is organized as follows. Chapter 2 introduces related work for SRAM aging analysis and aging mitigation methods. It furthermore summarizes the state-of-the-art of power, temperature and aging estimation in MPSoCs. Chapter 3 introduces the basics of reliability and gives an overview over aging mechanisms in nanometer technologies. Moreover, the effect of NBTI as a dominant aging mechanism is discussed. Chapter 4 summarizes the contributions of this work with regards to aging prediction and mitigation in on-chip SRAMs. First, the SRAM circuit design and operation are introduced, followed by a detailed discussion about the impact of aging on the individual components of the SRAM read-path. Afterwards, the newly developed application-aware aging analysis tool is introduced. Finally, two aging mitigation techniques are introduced and evaluated with the proposed reliability tool. Chapter 5 describes the work on runtime monitoring in FPGA prototypes and the implementation of the different monitors for power, temperature and aging. Chapter 6 presents experimental results for the new SRAM reliability tool and the two proposed aging mitigation schemes and evaluates the developed hardware monitors for an exemplary MPSoC architecture. Finally, conclusions are drawn in Chapter 7.

2 State of the Art

2.1 Aging Analysis and Mitigation Techniques for SRAMs

Aging analysis in SRAMs has been a subject of intensive research. However, most approaches concentrate on the analysis of SRAM core cells as well as on individual sub-components of the SRAM peripheral circuit. Consequently, many state-of-the-art approaches ignore the interaction of building blocks within the circuit and neglect the strong dependency of aging on realistic workloads. Similarly, state-of-the-art aging mitigation techniques focus mostly on aging mitigation within the SRAM cells and only few work has been published regarding aging mitigation in other critical building blocks like SAs. The following chapter reviews existing research work for aging analysis and mitigation in SRAMs and shows the contributions of this thesis as pointed out in Chapter 1.4 in comparison to the already established research work.

2.1.1 Aging Analysis of SRAMs

Previous studies have already thoroughly analyzed the impact of BTI on the read stability and static noise margin (SNM) of SRAM cells [15, 16]. [17] investigates the stability of an SRAM cell under worst-case conditions for both Negative and Positive BTI (NBTI and PBTI) appearing in PMOS and NMOS transistors, respectively. Not much work has been done regarding the aging characterization in SRAM peripheral circuits such as the write driver [18], timing control logic [19] and SAs [20]. [21] investigates the BTI impact on SRAM standard latch-type amplifiers for process, supply voltage and temperature variations. As indicated above, all of these works investigate individual sub-components of the SRAM disregarding the interaction with other building blocks.

Especially, the read operation is often considered as one of the most critical operations and can hence be regarded as one of the lifetime limiting factors in

SRAMs. [22] therefore investigates the impact of SRAM read-path aging while considering both cell and SA aging. However, the work uses artificial workloads, which do not represent realistic workloads from embedded applications and hence is not suitable to accurately predict the impact of aging on the read-path. Furthermore, aging effects in the timing control logic and signal drivers are neglected. [23] analyzes the impact of wear-out in all essential components of an industrial SRAM library but with artificial march stress patterns as workload which are used for memory testing. In contrast, the proposed AppAwareAge tool uses real workloads from embedded applications and considers aging in the complete read-path.

2.1.2 Aging Mitigation in SRAM Core Cells

Several mitigation techniques were suggested targeting cell aging optimization of SRAM-based register files inside CPUs. One category of these works applies bit flipping, which inverts stored bits in the register file after every write to achieve a duty cycle balancing [24–26]. These techniques usually are accompanied by a significant power overhead and more reads and writes since flipping the cell content requires to read out the corresponding cells and to write back the inverted bits. [27] introduces a recovery boosting technique which adds inverters to the SRAM cell to raise the storage node voltages to induce recovery in the PMOS devices of the cell. This method results in a significant power overhead and may, hence, not be applicable to large-sized memories.

Other work uses bit rotation methods to mitigate aging in the SRAM-based register files inside CPUs. The least significant bit is moved by one position in [28]. The work in [29] introduces a barrel shifter to rotate the assignment of register numbers of the register file using a bit count. Generally, such a barrel-shifter solution must be applied after the decoder stage because, otherwise, a rotate shift can be unsuccessful when all address bits of the non-decoded address are '0' or '1'. This is feasible because the register file is usually a relatively small SRAM memory. In contrast, this scheme cannot be applied to SRAM-based data memories. Due to the large width of the decoded address in bigger SRAMs the scheme would result in a huge barrel-shifter logic.

Another approach, in which memory addresses are re-assigned to different physical memory regions, is described in [30]. This work focuses on frame data in a video stream. For each frame, a new base address is calculated to move

the incoming frame to a new location in the memory. This is a very specialized memory management solution only applicable to mitigate aging in scratchpad memories for image/video processing architectures. In [31] an aging-aware coding scheme is proposed to balance the aging stress of SRAM cells with an architectural application simulator to obtain realistic workloads.

The works in [32–34] introduce techniques for balancing the duty factor of SRAM-based data caches by exploiting cache access characteristics. However, these methods are not applicable to SRAM-based on-chip data memories since they depend upon the inherent behavior of caches (such as flushing, cache hits etc.). In [35], a redundancy-based SRAM micro-architecture is used for extending the SRAM lifetime which requires the modification of the 6-Transistor (6T) SRAM cell.

2.1.3 Aging Mitigation in the Sense Amplifiers

Not much work has been proposed to mitigate aging in SRAM SAs. [9] introduces a mitigation technique for SA offset voltage degradation that balances the SA workload of each individual SA by modifying the SA circuit itself. It was analyzed with real workloads for an L1 data and instruction cache. In contrast, the proposed MAGIC approach applies a wear-leveling technique. [36] introduces a circuit-level approach called Logic-Wear-Leveling (LWL), which replicates critical and near-critical logic paths as well as switches between them to induce recovery times from aging. [22] proposes to mitigate aging in the SA and cell by increasing the drive strengths of the pull-down transistors. In contrast to the two mitigation techniques MAGIC and SDE proposed in this thesis, the work discussed in this section generally requires either a modification of the SA design or introduces a significant area/power overhead especially for large SRAMs.

Other research focuses only on the mitigation of the impact of process variations on the SAs. In [37] a tuneable SA design is presented to compensate in-die variations, while [38] monitors the offset voltage using an on-chip circuit to estimate yield.

2.2 Power, Temperature and Aging Emulation in Multi-Core Processors

Conventionally, software simulators are employed for power and thermal estimation which require extremely long simulation times, especially for full system simulations. To counter this, application snippets are employed and operating system effects are often ignored to reduce the runtime. Consequently, accuracy and credibility of the results may compromised. As an alternative, hardware runtime monitoring is a fast and accurate alternative for power and thermal estimation. However, this method is only applicable in existing processors. FPGA-based hardware emulation on the other hand provides the necessary flexibility and accuracy for power and thermal estimation in early design stages, while running at hardware speed. In the following, the state of the art of FPGA-based hardware emulation is introduced. Furthermore, the origin and contributions of this thesis to the State of the Art are highlighted.

2.2.1 Power Emulation

In [39] dedicated hardware accelerators for power models are mapped onto an FPGA. Each power module tracks input and output signals of different macro-blocks of the design. Speed-ups of 10 to 500 compared to simulation-based approaches are achieved, but resource requirements restrict the method to small designs. [40] proposes an FPGA-based emulation approach using event counters of processor components, which are evaluated in software-based component specific power equations. The method can provide up to 35x speedup compared to corresponding full-system software simulations. Similar approaches are presented in [41–43] where power monitors are created by using Performance Monitoring Units (PMUs) included in the processor. As in [40] PMUs include counters that can be programmed to capture events. The counter values can be read by software to conduct a performance analysis, which however does not represent a full hardware emulation and hence is not applicable for realtime monitoring. [44] introduces a combination of hybrid functional level power analysis (FLPA) and instruction level power analysis (ILPA). Gate-transfer level and power simulation of the processor allows to model power functions which can be implemented in the FPGA. The models enable the estimation of application-specific power consumption and energy per task. The approach in [45] discusses

an instruction-based energy estimation approach for off-line energy estimation by instruction set based simulations. The characterizations can be conducted up front and only few computations are needed at runtime which makes this approach applicable to real-time monitoring. [46, 47] proposed the evaluation of internal processor states (e.g., CPU idle, cache hit, memory write) which are monitored by dedicated power sensor units and used in linear regression based power models.

2.2.2 Temperature Emulation

Thermal-aware system emulation can be done off-line with detailed software simulators like in [48] or virtual platforms like in [49] which do not reach the speed of hardware-assisted emulation. Some related work on off-line detailed simulations can be adapted for real-time temperature monitor emulations. [48] therefore discusses a computationally intensive thermal RC modeling approach for simulating the temperatures of processors which will be utilize in this work to generate a temperature model that can be mapped onto the FPGA and hence used for temperature emulation. Other work enhances this approach regarding computation costs [50]. Another approach proposes techniques that perform FPGA emulation of the functionality but utilize an external thermal software model running on a host computer to estimate the power consumption and thermal behavior of each processor core [51]. However, the external simulation does not represent a complete hardware emulation, limiting the performance of the approach. [52] proposes thermal exploration during runtime relying on existing on-chip temperature sensors which might either not be available or can only represent the temperature of the prototype, not the ASIC implementation. [53] proposes an online simulation approach which is based on executing thermal simulation as a software task on multiprocessors. [54] proposes a thermal-aware system emulation framework which employs the thermal model as differential equations to calculate the current core temperature.

2.2.3 Aging Emulation

Aging of logic has been extensively analyzed [55, 56]. However, not much work has been presented on the hardware-assisted emulation of aging in FPGA Prototypes. [57] introduces an online monitoring approach of representative critical

gates. It employs a workload monitor that observes a subset of the circuit's inputs, and a temperature sensor. Based on the state of the inputs, the workload monitor predicts the current stress of each representative critical gate. In combination with a software component the degradation rate is predicted based on the aggregated output of the workload monitor and the temperature sensor. Even though this approach has only been evaluated on very small digital circuits it could potentially be deployed for the aging emulation of FPGA Prototypes of MPSoCs, especially if combined with a temperature emulation approach, instead of a temperature sensor.

2.2.4 Enhancement of the State of the Art

The proposed FPGA emulation method in this thesis links to [14], where the power is emulated with an instruction-level energy model, since compared to the event-based models fewer computing operations are required during run-time. The method however neglects the power consumption of data and instruction caches or frequency-voltage scaling. Temperature emulation is achieved with the modeling approach from [48]. Although its thermal RC model is computationally very intensive, the simulation time to obtain model data is not critical, since offline simulations are performed only once up front, not during emulation. The model furthermore shows good agreement with finite-element simulators and test-chip results. The work in [14], however evaluates no real tasks or task distributions from the running prototype. The temperature values are stored in a LUT and the effect of neighboring cores is modeled with the addition of a fixed temperature value to the single-core temperatures for each active neighbor, which is not accurate since the temperature behavior is not linear. Aging is not monitored in this work. To the best of our knowledge, the approach presented in this thesis is the first work towards a full system real-time monitoring system for FPGA prototyping of single- or multi-processor systems, that emulates data characterized from the target ASIC design and includes not only power and temperature, but also aging monitors.

3 Reliability and Degradation

While technology scaling has enabled higher on-chip integration densities it has also introduced problems like larger internal electric fields since the feature size is scaled more aggressively than the supply voltage in order to avoid a significant increase in leakage current. Elevated fields increase the impact of transistor wear-out and induce time-dependent parameter drifts which cause a degradation of the electrical properties of a transistor over time. Transistor performance gradually decreases compared to its fresh state right after manufacturing hence compromising IC reliability. The trend of an increasing power consumption naturally leads to higher on-chip temperatures, which aggravate transistor aging mechanisms.

In today's manifold systems, the requirement that a system is free from defects and systematic failures at time $t = 0$ is hence not sufficient any more. Rather, nowadays systems are expected to provide a failure-free and fail-safe behavior for a stated time interval usually denoted as the lifetime of a system. Quantitatively, reliability is defined as the probability that a product will perform its required function under given operating conditions for a specified period of time [58]. Given, that the systems performs correctly at time 0, the reliability R(t) defines the probability that a system functions without any failure during the time range $[0, t]$. It continuously measures a system's capability of delivering correct service. Hence, reliability is inherently a time dependent function and longer operation will lead to a reduced system reliability.

With rising system complexities and rapidly growing costs induced by loss of operation as a consequence of failures, reliability excelled as a new major design goal. To ensure an operation with an acceptable error rate and prohibit a failure of the product before reaching its intended lifetime, excessive investments are necessary to maintain the circuit reliability. Nowadays, the increasing reliability costs (in terms of power, area, design cost, etc.) tend to compensate the performance gain obtained by moving from one technology to the next (see Fig. 3.1). Consequently, a transition to the next technology node might no longer be profitable. For that reason, also the research effort on aging analysis method-

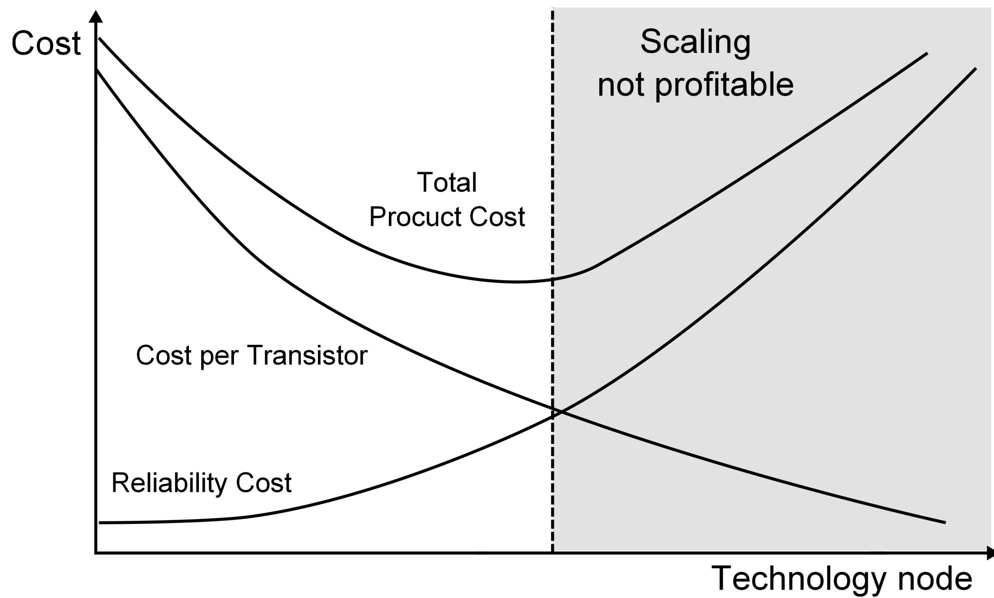

Figure 3.1: Development of reliability/variability and transistor costs for the different technology nodes [61]

ologies to accurately analyze the performance degradation of aging circuits has greatly increased [59, 60]. This chapter introduces the basic concepts of reliability and reliability modeling and provides an overview of the aging mechanisms in nanometer technologies. The remainder of this chapter is organized as follows. Firstly, the failure rate curve and its importance for reliability analysis is discussed in Section 3.1. Next, the concept of device physics of failure is introduced and the most dominant aging mechanisms in nanometer technologies are discussed in Section 3.2.

3.1 Failure Rate Curve

The failure rate plays an important role in reliability analysis. The reliability of a population of semiconductor devices is represented by the failure rate curve, which is commonly known as the bathtub curve. The failure rate is the ratio of the items which failed within a given time interval compared to the number of items still working. The corresponding curve shown in Figure 3.2 reveals that the operation of semiconductor devices is comprised of three distinct periods: 1) infant mortality region: early failures which occur within a short time after

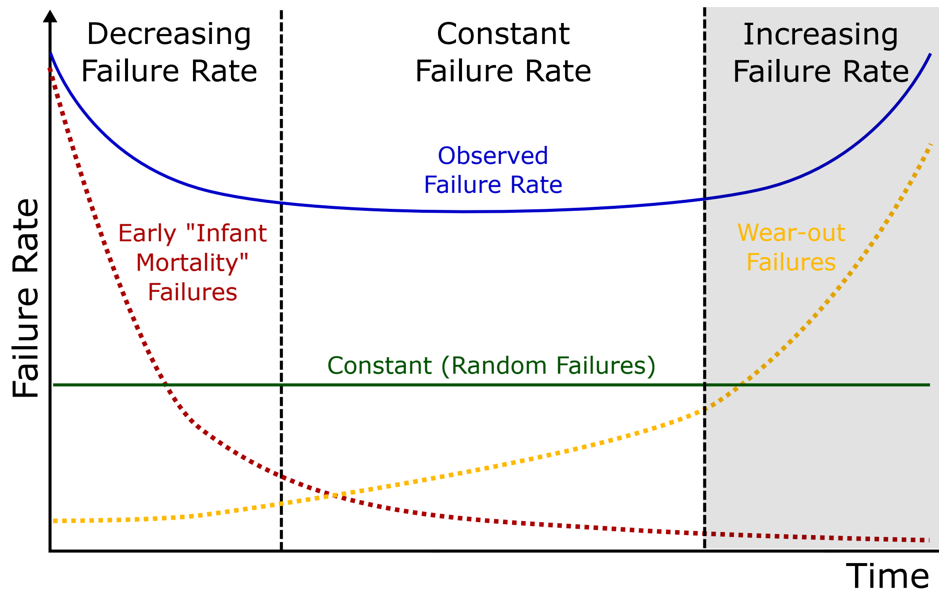

Figure 3.2: Failure Rate Curve (Bathtub Curve)

device operation is initiated, 2) useful operating life: random failures over time with almost constant failure rate, 3) system wear-out region: wear-out failures with increasing failure rate as the device approaches its end of life [62].

Early failures are caused by devices which inherently contain defects due to variations in the manufacturing equipment, device dimensions, the presence of particles etc. [63]. The failure rate in the early failure period decreases with time because failures manifest themselves within a short time after operation under temperature and voltage stress. To reduce the failure rate in this period and improve quality, typically a screening process is performed, which can include stress, burn-in and other types of electrical characteristics testing. This process ensures that most of the weak devices fail already at this stage and therefore never reach the customer. Once products with defects are removed from the lot, only products with a low failure probability are left. The rate of good products obtained after screening is called yield. Products with higher yield consequently have a lower defect density.

The useful operating region has two contributors. After screening out devices which fail due to production weaknesses, products with minor defects remain which operated stable during screening. Therefore, the random failure region starts as a continuation of the early failure region with an attenuating failure rate over time. At the same time, random failures occur with a constant failure

rate which are not caused by defects in the product but by external conditions like electrical noise, soft errors, overvoltage, electrostatic discharge etc.

The final stage in a device life cycle is the system wear-out region. Wear-out failures can be attributed to degradation mechanisms and fatigue. Even though degradation occurs right after production its impact will grow significantly with increasing time until the failure-rate drastically rises and the product is not fit for use any more. The entering point of the wear-out region depends on the operating conditions and the product itself.

To predict the impact on device reliability at this last stage of the failure rate curve, reliability analysis is deployed to investigate the ultimate resulting performance for a given set of constraints like e.g. design rules, operating voltage, temperature and maximum switching speed to determine the development of the failure-rate and predict how long a given product can be guaranteed to operate safe and failure-free [64]. Accordingly, reliability modeling for the purpose of lifetime prediction and to estimate the susceptibility of an IC design to failure mechanisms has become indispensable and steadily needs to be evolved.

Reliability modeling is nowadays mostly based on the physics-of-failure concept which leverages the knowledge of the root cause and the physical behavior of the key failure mechanism from which analytical models that can forecast the electrical device characteristics in the wear-out region are established. Reliability simulation tools make use of these mathematical models which are often integrated into the circuit simulator to analyze and predict product reliability and improve product performance. The design and implementation of such reliability tools is becoming especially critical for the deployment of design-for-reliability (DfR) techniques which allow IC designers to address reliability concerns already at the design stage. The useful deployment of reliability simulation tools in DfR can reduce the number of iterations in the conventional and costly design-test-redesign cycle [65]. Hence, DfR strategies decrease time-to-market and development costs while ensuring that product design is optimized for reliability before moving to the manufacturing step.

In the following sections, the most important degradation mechanisms are discussed. Since BTI has been identified as one of the major reasons for transistor wear-out, the mathematical physics-of-failure-based models for BTI, which are utilized in this work, are introduced.

3.2 Electron Device Physics of Failure

Transistor aging has been observed for several decades but only started to cause serious issues in advanced technology nodes where time-dependent variations have reached a crucial level and excessive margins are necessary to maintain failure-free operation. The main transistor wear-out mechanisms are Bias Temperature Instability (BTI), Hot Carrier Injection (HCI), Time-Dependent Dieletric Breakdown (TDDB) which can all be related to defects in the oxide. Another wear-out mechanism is Electromigration (EM), which causes the migration of metal atoms in a conductor due to an electrical current and severely impacts the reliability of metal interconnects. BTI and HCI gradually shift the performance parameters of the affected transistors and can be counteracted well through extra design margins. TDDB on the other hand slowly establishes a conductive path through the gate oxide which eventually permanently damages the transistor and is therefore better handled by Mean-Time-To-Failure Analysis [66].

3.2.1 Negative Bias Temperature Instability

One of the most important degradation mechanisms in modern CMOS technologies is Bias Temperature Instability (BTI). BTI results from trap generation inside the gate oxide or the interface layers between the gate oxide and the substrate [67]. The prevalent models attribute BTI either to a Reaction-Diffusion process [68] or atomic-scale mechanisms based on the capture and emission of traps in the oxide [69]. The Reaction-Diffusion model used in this work describes BTI as an electrochemical reaction-diffusion process containing two distinct phases: Transistors are aging while under BTI stress and recover whenever the stress is removed. This is shown in Fig. 3.3 (short-term aging) where the transistor threshold voltage gradually increases during the stress phase and partly recovers during the recovery phase. For PMOS devices, this effect is known as Negative Bias Temperature Instability (NBTI). It is explained in detail in the following.

The BTI stress phase occurs while the transistor operates in the triode region, i.e. at high $|V_{GS}|$ ($V_{GS} \approx -V_{DD}$) and very low V_{DS} ($V_{DS} \approx 0$). In the triode region the whole channel is in inversion and can therefore contribute to charge trapping. Hence, defects at any location in the oxide can trap holes from the channel and a captured positively charged defect increases the transistor threshold voltage.

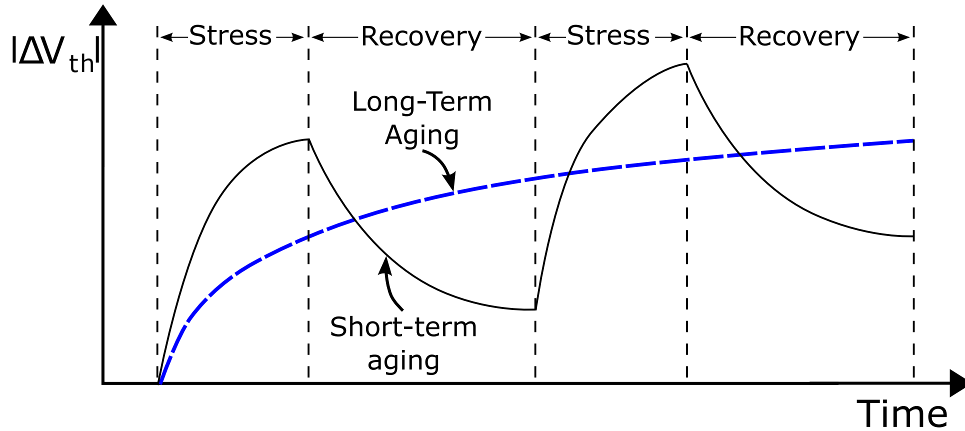

Figure 3.3: BTI-induced V_{th} shift during stress and recovery

During the stress phase, holes in the inversion layer interact with $Si-H$-bonds at the $Si-SiO_2$ interface of the transistor gate due to the influence of the strong vertical electrical fields. This interaction can break the $Si-H$-bonds, subsequently liberating H/H_2 species which diffuse away from the $Si-Si0_2$ interface towards the poly gate or anneal to other existing traps as shown in Fig. 3.4. As a consequence, dangling silicon atoms are left at the $Si-SiO_2$ interface of the transistor gate creating traps which are often referred to as interface traps that may become activated. Activated traps gradually increase the threshold voltage (V_{th}) of a transistor, which results in a poorer drive current and lower noise margins.

During the recovery phase, escaped H species start to diffuse back to the $Si-SiO_2$ interface and recombine with the Si species to $Si-H$ bonds as shown in Figure 3.5. Removal of the stress can only anneal some of the interface traps resulting in a partial recovery only. Hence, the long-term effect of BTI is a shift in the threshold voltage of the transistor as shown in Fig. 3.3 (long-term aging - responsible for degradation) [70]. Since BTI arises from defects inside the gate oxide or at the gate/substrate interface, its characteristics strongly depend on the process technology and the material and thickness of the gate oxides which need to be accurately captured in a mathematical model.

3.2.2 Positive Bias Temperature Instability

The corresponding phenomenon in NMOS transistors is called Positive Bias Temperature Instability (PBTI). Similar to NBTI, it originates from charge trap-

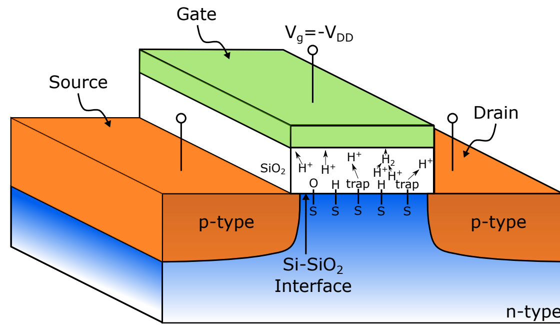

Figure 3.4: NBTI - stress phase

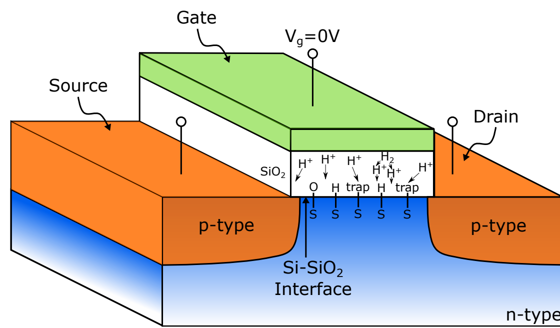

Figure 3.5: NBTI - recovery phase

ping inside the gate oxide or at the gate/substrate interface layers. With the introduction of high-k metals PBTI has emerged as a serious reliability threat and can no longer be ignored [71]. High-k oxides and metal gates are introduced to reduce gate leakage currents and therefore enable further supply voltage scaling. However, although the vertical field decreases, more traps are available to be charged.

3.2.3 Analytical Bias Temperature Instability (BTI) Models

Two distinct BTI models will be used in this work, namely a high and a low accuracy model. The high accuracy model is utilized for very precise estimations of the aging behavior in on-chip SRAMs. The low accuracy model is used in the hardware monitoring approach for FPGA prototypes due to its reduced complexity.

BTI Model with High Accuracy

To capture the aging degradation due to NBTI and PBTI with very high accuracy the long-term model from [67] is utilized. The model captures the recovery effect and the dependency on the process technology, material and oxide thickness of the gate and hence enables a very precise estimation of the V_{th}-shift. Analytically, the V_{th}-shift can be accurately described over time with the following equation:

$$\Delta V_{th}(t) = A \left(\frac{\sqrt{K_v^2 T_{clk} \gamma}}{1 - \beta(t)^{1/2n}} \right)^{2n} \quad \text{with} \quad K_v = f(V_{dd} - V_{th}) \tag{3.1}$$

where T_{clk} is the clock period, K_v is a function of the supply voltage and the parameter A is used to fit the model to the results in [71]. γ determines the fraction of stress and recovery that a specific transistor inside the SRAM experiences or in other words its workload since not all transistors are equally under stress. All other parameters are technology-dependent parameters and are described in [67].

BTI Model with Low Accuracy

To reduce the complexity of the equation and hence the hardware overhead in the FPGA design, the model from S. Nassif (personal communication, November 10, 2016) is utilized. The model neglects some important technology and design dependencies but provides the benefit, that the equation can be realized with less hardware overhead. This model will be used for the FPGA realization of the aging monitor.

$$\Delta V_{th}(chipage) = 0.05 \cdot exp(\frac{-1500}{T_c}) \cdot {V_{DD}}^4 \cdot chipage^{\frac{1}{6}} \cdot \alpha^{\frac{1}{6}}$$

where T_c equals the current core temperature in K, V_{DD} the current supply voltage, $chipage$ the lifespan (age) of the chip in seconds and α the transistor workload, i.e. the the fraction of stress and recovery.

3.2.4 Other Degradation Mechanisms

Hot Carrier Injection

Despite the severeness of BTI degradation in present technology nodes, other effects such as hot carrier injection (HCI) should also be considered as secondary effects. HCI damages the transistor due to accelerated carriers and acts on both nMOS and pMOS transistors. The term "hot" refers to the required high kinetic energy and hence the carrier velocity. In NMOS transistors, electrons are accelerated in the lateral electric field and gain kinetic energy. Near the drain end, the carrier energy might be high enough to overcome the potential barrier between the silicon and the gate oxide. Some carriers can leave the channel and get injected in the gate oxide or collide with silicon atoms within the substrate. This collision might initiate impact ionization where high-energetic electrons are generated which are attracted towards the gate-oxide. As a result, some of these 'lucky electrons' may become trapped in the gate substrate interface [72, 73]. A threshold voltage increase is induced by the small portion of carriers which are caught in the gate oxide. Consequently, a degradation of the drain current deteriorates the transistor performance. In PMOS transistors a similar mechanism is induced by "hot" holes. Due to the lower mobility of holes, HCI is more critical in NMOS devices. Depending on the stress conditions there are four distinguished injection mechanisms for the injection of hot carriers into the dielectric: channel hot-electron (CHE) injection, drain avalanche hot-carrier (DAHC) injection, secondary generated hot-electron (SGHE) injection, and substrate hot-electron (SHE) injection. HCI gets more severe for stronger electrical fields resulting from shorter channel lengths and, higher drain-source voltages and reduced oxide thicknesses. Contrary to NBTI, HCI degradation requires a current flowing and thus occurs during transitions in between logic states. Consequently, HCI gets more critical for higher operating frequencies when transistor are switching more often. Although HCI is negligible in the fast input regime, it becomes nearly as severe as NBTI for slow input slew rates. Furthermore, HCI degradation strongly depends on the signal slope described by parameters like the slew rate or the fan-out [74]. Although the impact of HCI is well recognized, this work will only focus on BTI modeling and simulation but the framework can easily be enhanced to include HCI aging.

Time-dependent Dielectric Breakdown and Electromigration

Further reliability concerns are caused by degradation mechanisms like Time-dependent dielectric breakdown (TDDB) and Electromigration (EM). TDDB occurs when traps in the oxide start to form a conductive path through the gate oxide to the substrate which is referred to as soft breakdown. A hard breakdown occurs once the conductive path is fully established which in turn ultimately heats up the material and hence allows for increased conductance until the silicon starts to melt. The melted material forms a silicon filament which permanently damages the transistor [75]. Electromigration is caused through the transport of material in an electrical field. Current flow through a conductor produces a force generated by the momentum transfer between conduction electrons and metal ions in the crystal lattice.

3.3 Summary

This Chapter introduced the basic concepts of reliability and reliability modeling. Moreover, the root cause and physical behavior of the most dominant aging mechanisms in advanced technology nodes were discussed. Two distinct analytical models to represent the threshold voltage shift caused by NBTI and PBTI are introduced in detail.

It should be noted that to achieve a high level of device reliability also at high temperatures, all aging effects should to be taken into consideration. Especially, since an interaction between different aging effects can generally be observed where e.g. the delay degradations due to HCI and NBTI can either add up or even compensate one another. This partial compensation induces that the impact of device degradation on the circuit performance cannot be generally predicted but is strongly dependent on operating and workload conditions and hence has to be investigated individually for each design [59]. It should be stated that for a comprehensive reliability study of a circuit, all aging effects need to be considered. This should especially be kept in mind if highly-reliable circuits in safety critical systems are the object of investigation. Although the impact of other aging mechanisms is well recognized, this work will only focus on BTI modeling and simulation. The proposed reliability tool however is designed such that it can easily be extended to include other degradation mechanisms.

Despite of the known existence of these reliability challenges there is still a lack of sophisticated analysis methods for the design phase, especially with respect to an accurate integration of realistic workload scenarios. This highlights the need for new accurate reliability simulation tools, which guarantee reliable operation and correctly estimate necessary guardbands.

4 Application-Aware Aging Analysis and Mitigation for on-Chip SRAM Design-for-Reliability

Many embedded systems such as automotive and industrial Micro-Controller Units (MCUs) use on-chip Static Random Access Memories (SRAMs) as main data memory. In addition, SRAMs constitute a significant portion of most embedded systems' chip area. However, SRAMs are especially vulnerable to wear-out mechanisms such as Bias Temperature Instability (BTI) since they are designed for highest integration densities and continue to lead the migration to new technology nodes [76]. BTI leads to a significant performance degradation, especially in the Sense Amplifiers (SAs) within the SRAM read path which are crucial for high performance [77]. At the same time, the failure of an SA is particularly critical. It destroys the read-out of the whole column which is multiplexed to the failing SA and renders the data of every cell in that column useless, which usually marks the end of the MCU's lifetime

To compensate for aging, designers usually introduce guardbands by adding extra design margins to the circuit to guarantee proper functionality throughout its specified lifetime. These guardbands need to be stacked on top of other guardbands, e.g., to compensate for the increasing impact of process variations. Hence, additional guardbands should be narrowed down as much as possible, which requires accurate aging prediction for a given SRAM architecture and device technology. Since BTI aging affects all active devices, an accurate aging prediction requires to take into account all SRAM components. An isolated analysis of the individual components may result in optimistic or pessimistic results [23]. Additionally, aging design margins are typically based on worst-case workload scenarios, which rarely match the workload induced by real applications [9]. Accurate aging prediction however should better be based on realistic workload information. Otherwise, pessimistic assumptions lead to very wasteful margins in terms of area and power overheads as well as speed penalties. This is especially critical for systems, which are designed for a long lifetime such as automotive and industrial MCUs.

To address these challenges, a novel reliability tool for SRAM Design-for-Reliability is introduced as first contribution of this thesis. The tool AppAwareAge incorporates an application-aware aging analysis for on-chip SRAM data memories. It enables a precise analysis of the memory aging considering the workload caused by an application executed on the MCU using application-specific memory traces. The focus of this method lies on the accurate aging prediction of the complete read-path with all its components. The method incorporates aging of SRAM cells, aging of SAs as well as aging of the SAs' pre-charge circuitry and signal drivers along the read-path. We investigate both the degradation of the sensing delay (which reflects SA aging) as well as the degradation of the bit line swing (which reflects SRAM cell aging). In addition, the analysis method is extended to predict the expected end-of-life of the SRAM. We demonstrate the method by analyzing SA and SRAM cell aging for various workloads, temperatures and supply voltages. The presented results show (1) that realistic workloads are an extremely important factor for the accurate prediction of aging, (2) that the aging of SAs was often overestimated in previous work [22,78], (3) that, depending on the cell sizing, both SA and SRAM cell aging have a significant contribution to the degradation of the read-path and (4) that, non-intuitively, aging in the SA's control signals counteracts the bit line swing degradation caused by the cell aging to some extent leading to minor performance improvements. Detailed analysis shows the cause of this positive aging effect.

Next to guardbands, mitigation schemes can be applied to counteract aging. Such mitigation schemes can avoid unbalanced aging [78] and significantly reduce design margins. The results of our workload-aware aging analysis show that certain address ranges of on-chip SRAM data memories are accessed much more frequently than others for the investigated workloads. These sections correspond to frequently used data sections and the stack. This leads to an exacerbated aging of the corresponding SAs and, hence, necessitates larger design margins.

As countermeasure, we propose as next contribution of this thesis two distinct aging mitigation techniques to mitigate aging in the SAs of SRAMs. The first mitigation technique, called the Mitigation of AGIng Circuitry (MAGIC) is an extremely simple circuit that modifies the mapping of SRAM banks to physical addresses in order to distribute the memory accesses evenly over the complete SRAM array. It is, to our knowledge, the first hardware-based wear-leveling circuit suitable for on-chip SRAM data memories. With MAGIC, stress is evenly distributed between SAs, thereby sparing them from exaggerated aging stress.

We apply the proposed application-aware aging analysis to compare the aging behavior of the read-path with and without MAGIC. We demonstrate its effectiveness for various workloads, temperatures and supply voltages. MAGIC can mitigate SA degradation by up to 26% for 3 years of aging while showing a minimal area and performance penalty. Furthermore, it is shown that this results in an extension of SRAM lifetime of 3x for harsh operating conditions.

For the second mitigation technique the proposed aging analysis AppAwareAge is integrated into an aging-aware SRAM design exploration framework (SDE) that generates and characterizes memories of different array granularity (e.g. number of banks/rows/words). The presented results show that for an intended set of applications, aging rates can differ significantly for different memory architectures hence making some architectures more reliable than others. It is shown, that selecting the most reliable memory architecture in terms of aging can mitigate SA degradation significantly depending on the environmental conditions and the application workload. SDE can improve SA degradation by up to 31.6% for 3 years of aging while having a low area penalty. Exploring possible array configurations early in the design phase can significantly mitigate SA aging and hence improve the memory lifetime.

The remainder of this chapter is organized as follows. Firstly, the basic SRAM circuit design and operation is introduced in Section 4.1 while Section 4.2 gives an overview over important performance parameters. In Section 4.3 considered aging mechanisms and their impact on the SRAM read-path are discussed. Section 4.4 introduces the proposed application-aware aging analysis AppAwareAge. Afterwards, the two mitigation techniques MAGIC and SDE, which balance the workload and mitigate aging of SAs, are described in detail in sections 4.5.1 and 4.5.2, respectively. Finally, the chapter is concluded in Section 4.6.

4.1 SRAM Circuit Design and Operation

Apart from the introduction of hierarchical memory to decrease memory access latency and prevent processors from hitting the memory wall, where memory access times become the limiting factor of system performance [79], another trend is the increasing level of integration of complex electronic components and functions into a single silicon chip. Powered by the advances of technology scaling, these System-on-Chips (SoCs) contain pre-designed and pre-verified intellectual

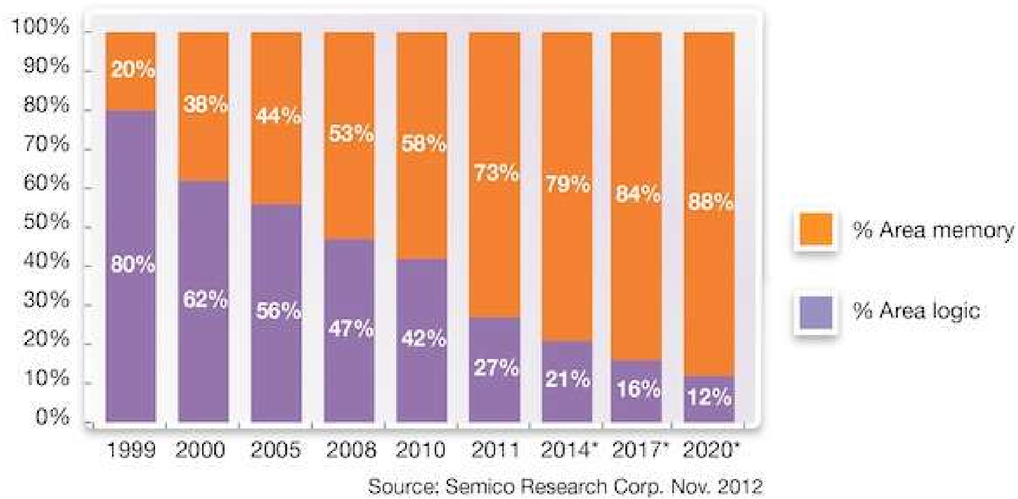

Figure 4.1: The trend of embedded memory content in high-end SoCs

property (IP) blocks like embedded processors, interface blocks, analog and digital blocks and components that handle application specific processing functions which used to be integrated on a board as separate chips. SoCs feature various benefits such as shorter design times, lower overall cost, smaller form factors and reduced power consumption and hence have found wide application from consumer electronics like smart phones to high-end and mobile computing [80].

Another important family of building blocks in SoCs is on-chip memory which allows to support the increasing need for data bandwidth while decreasing the power consumption compared to off-chip memories. Today's feature-rich applications require the ability to store and access increasing amounts of data. The execution of most instructions involves one or more accesses to the memory inevitably leading to an explosion of the memory requirements of application programs. Hence, SoC performance depends heavily on the storage capacity and access speed of memories. Because of this increase in memory demand, memory blocks on SoCs nowadays claim most of the chip die area and SoCs have transitioned from logic-dominant to memory-dominant devices [66]. As shown in Fig. 4.1, memory blocks already account for almost 90% of the die area and their share is still projected to increase [3]. Among the different non-volatile memory technologies that are suitable to be embedded in SoCs, SRAMs are prevalent as on-chip memory in SoCs because of their compatibility with the CMOS logic process technology and voltage levels. Compared to other memory technologies, the fast access times of on-chip SRAMs allows them to close the

processor-memory performance gap. In comparison, DRAMs are typically used as off-chip main memory, where access speed is less critical and higher storage capacity is needed. Apart from the comparatively low access speeds, the manufacturing process of DRAMs is not compatible with CMOS logic and hence complicates on-chip integration. Flash memories are also well-suited for on-chip integration. However, they do not offer enough endurance as the number of program-erase cycles is limited since programming and erasing the memory damages the oxide layer and compromises reliability. Commonly, this issue is addressed with wear-leveling approaches which count the number of writes and dynamically remap memory blocks to guarantee a uniform utilization of the different blocks. Given the fact, that flash can only be erased block-wise, erase cycles are slow, giving it a significant speed disadvantage compared to SRAM. Other memory technologies like Ferroelectric RAMs, Phase Change Memories, Magnetoresistive RAMs and Resistive RAMs are not yet able to compete with existing memory technologies, since production costs are still high and other critical parameters like write-endurance, scalability and dynamic power consumption are not yet superior to SRAMs [66]. Hence, for the foreseeable future, SRAMs will continue to be the predominant memory in embedded systems.

4.1.1 SRAM Architecture

Fig. 4.2 shows an overview of the basic block structure of an SRAM array. The SRAM core array consists of 6T SRAM cells. The cells are arranged in K banks with a size of IxJ each, where I is the number of rows and J is the number of words. Hence, $(J * L)$ is the number of bits stored in a row for a word length of L. An address bus delivers the address to the SRAM, where the address is split into three address chunks: the bank address, the row address and the word address. A bank decoder receives the bank address from the address bus, decodes the address and selects the accessed bank according to the incoming address. A row decoder consequently decodes the incoming row address bits and enables the word line of the accessed word. In a word-oriented SRAM, a column decoder selects which bit lines are accessed and multiplexes the selected word to L SAs. To save area, the multiplexer permits that all bit line pairs of one word share the same set of sense amplifiers. For bit- or byte-wise addressable memories the number of sense amplifiers decreases accordingly since each bit- or byte is sharing the same set of SAs.

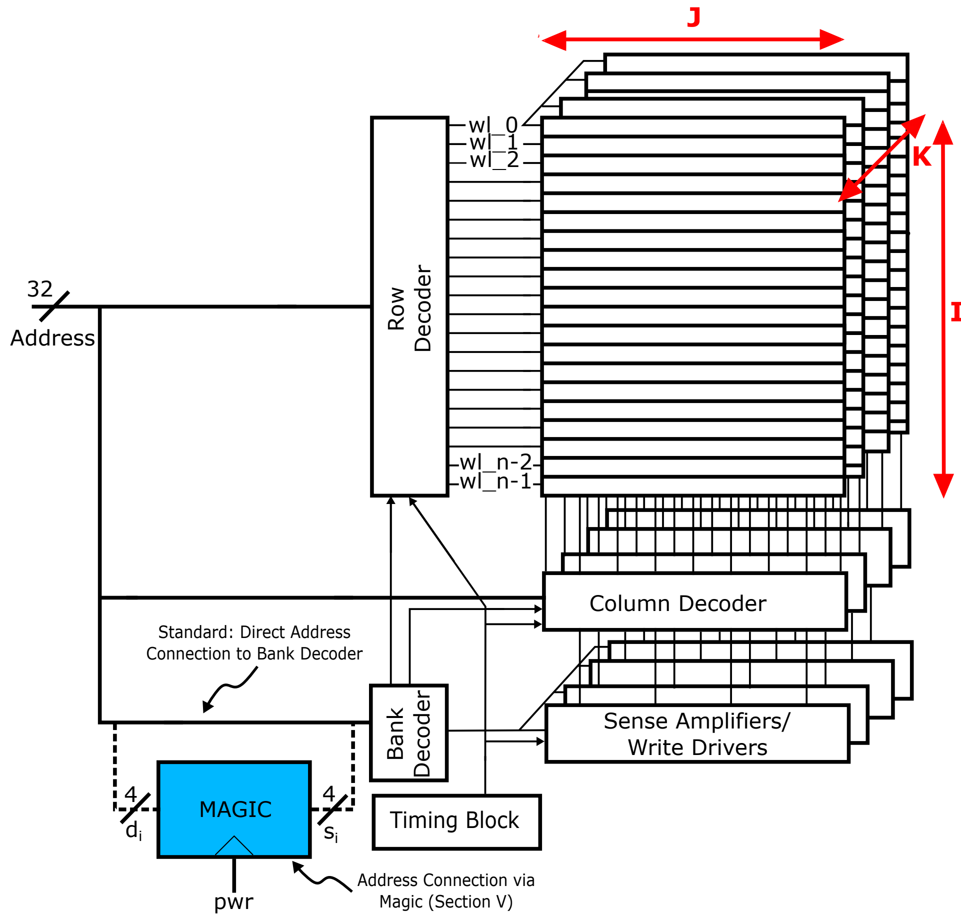

Figure 4.2: SRAM Block Architecture [81]. To mitigate aging, the blue box labeled *MAGIC* will be inserted in front of Bank Decoder (See Sec. 4.5.1), the box is not present in the standard SRAM architecture.

In the standard configuration, a bank decoder directly receives the address from the address bus and selects which bank to access. In Section 4.5.1, this flow will be changed by inserting an additional circuit block (blue box) that implements the Mitigation of AGIng Circuitry (MAGIC) before the bank decoder. Besides, each SRAM architecture contains a timing block which receives signals like the clock, chip select (in multi-SRAM chip architectures) and other control signals. The majority of SRAMs is self-timed, that means the timing control uses a replica bit line to generate all internal timing signals to ensure the correct timing of the SRAM.

Figure 4.3 shows the cross-section of the SRAM read-path, which consists of 6T-SRAM cells arranged in columns and a read-out circuitry. The cells are con-

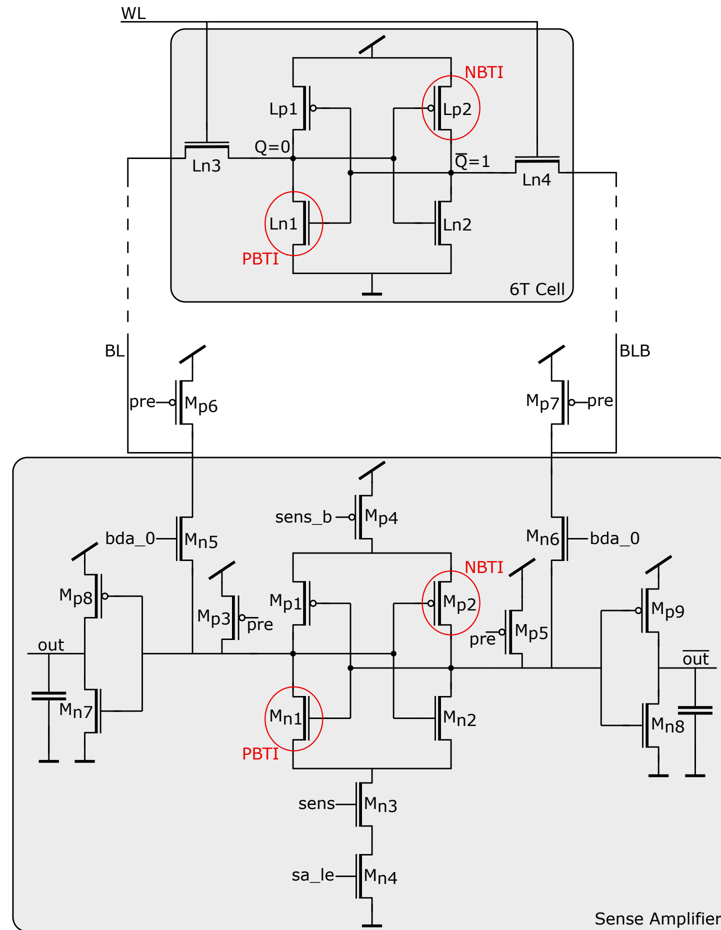

Figure 4.3: Cross-section of the SRAM read-path

nected via the bit lines BL and BLB. The cross-section shows only one of $(J * L)$ columns of the array, only the topmost of the I SRAM cells as well as only one SA. Each bit line possesses a global bit-line pre-charge logic to charge the bit lines to V_{DD}. The read-out circuitry uses a latch-type voltage SA. Each SA includes its own pre-charge logic to equalize the SA inputs. The signal control circuitry is not shown in the figure. In the following sections, the operation of the main SRAM building blocks will be explained in detail for read, write and hold operation of the SRAM.

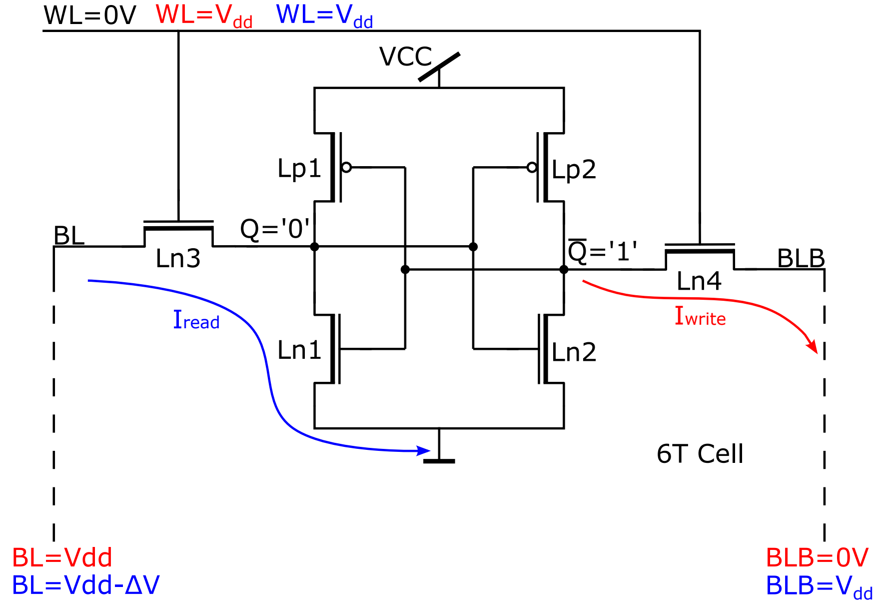

Figure 4.4: 6-Transistor SRAM Core Cell

4.1.2 Circuit Design and Operation of the 6T-SRAM Core Cell

The most common SRAM cell design uses a bi-stable flip-flop consisting of six transistors. This structure is called the 6-Transistor (6T) core cell and is depicted in Fig. 4.4. The bistable flip-flop is comprised of two cross-coupled inverters (L_{n1}–L_{p1}, L_{n2}–L_{p2}) which are connected to the bit lines via two access transistors (L_{n3}, L_{n4}). If the word line WL is activated, the internal nodes Q and $\bar{Q}$ are connected to the true bit line BL and the complementary bit line BLB, respectively. The complementary voltage levels at the internal nodes represent the stored bit.

Although there is a variety of other SRAM cell designs, the 6T cell topology is the preferred design, since it provides a superior trade-off between robustness, low-power and low voltage operation and area, especially for larger on-chip memories [81]. Cell designs with 4 transistors (4T) are attractive to reduce cell area but generally show a reduced stability and higher standby currents. Several cell topologies with a higher number of transistors have been proposed which sacrifice area to improve memory characteristics like reliability or power efficiency. Among them, mainly the 8T cell has received attention due to its advantages regarding the read stability.

The SRAM operation for the three modes hold, write and read of the 6T SRAM core cell used in this work are explained in detail in the following.

Hold During hold mode, the word line is driven low ($WL = 0V$) and hence the access transistors (L_{n3}, L_{n4}) are disabled. Due to the internal feedback, the cross-coupled inverter pair retains the stored bit as long as the power supply is on.

Write The write operation starts by connecting the bit lines to a write driver, which drives one of the bit lines to V_{dd} (in this example BL) and the other to ground, depending on the data input. Next, the word line is driven high ($WL = V_{dd}$) to enable the access transistors and establish the connection to the bit lines. The internal node storing the value '1' (in this example $\bar{Q}$) is discharged through the access transistor (L_{n4}) until the node $\bar{Q}$ is pulled below the trip-point of the inverter L_{n1}–L_{p1} and the cell state is flipped.

Read Prior to a read operation, the bit lines are precharged to V_{dd}. Next the word line is enabled ($WL = V_{dd}$) to establish the connection between the bit lines and the internal nodes of the cell. The bit line that is connected to the node which is storing the value '0' (in this example Q) consequently discharges through the path formed by the access transistor (L_{n3}) and the pull-down transistor (L_{n1}). Since one bit line stays pre-charged and one bit line discharges, a small voltage difference is generated which can be sensed by the sense amplifier.

4.1.3 Circuit Design and Operation of the Voltage-Latch Sense Amplifier

After the read operation has been initiated in the core cell, one of the bit lines discharges hence generating a voltage difference between BL and BLB. This small differential voltage called the bit line swing (BS) needs to be sensed and transformed into a full-swing output signal. The use of sense amplifiers greatly improves the power consumption and speed of the SRAM, since the bit line does not need to be fully discharged. There exists a large variety of sense amplifiers and the choice and design of a sense amplifier depends on trading off partly contradicting design constraints. In the following, a common Voltage Latch-Type SA design as shown in Fig. 4.5 is introduced to explain the detailed operation of the bit line sensing. The design was chosen because of its high speed advantage. Just like the 6T SRAM cell, it consists of two cross-coupled inverters (M_{n1}–M_{p1}, M_{n2}–M_{p2}) whose internal nodes (P and $\bar{P}$) are connected to the input of the

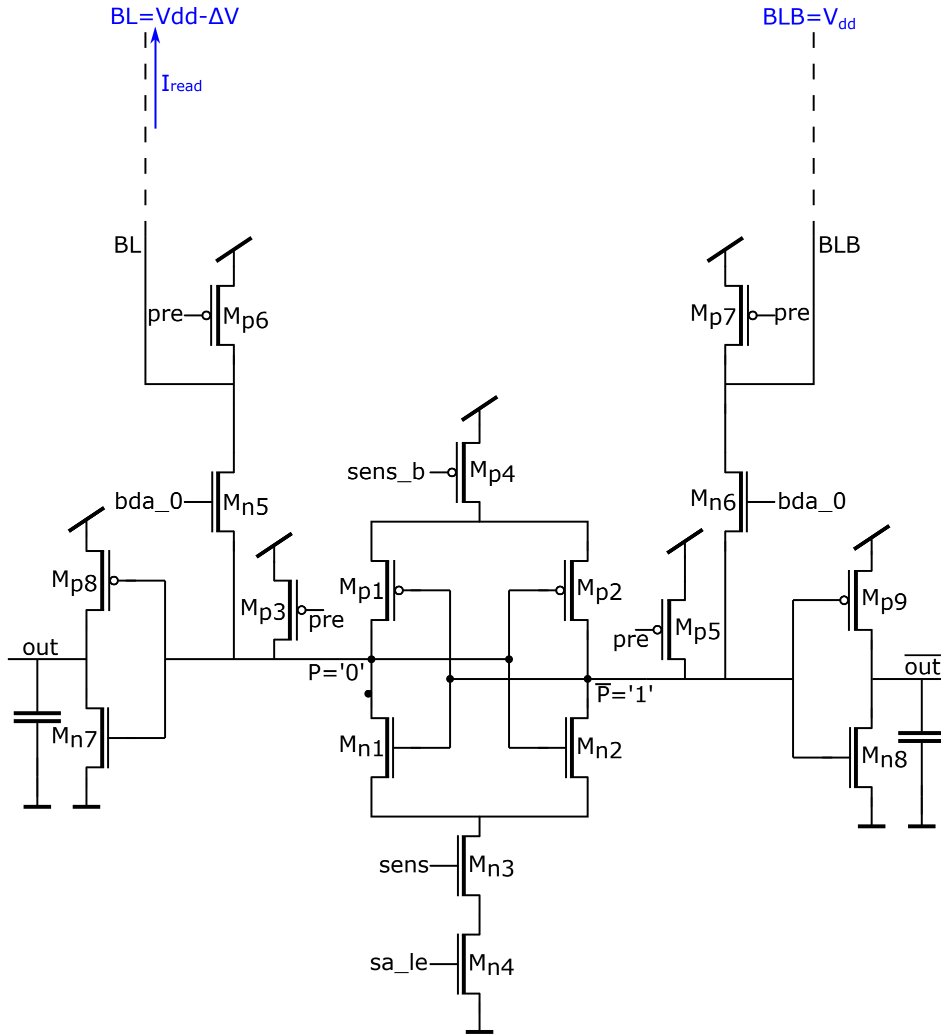

Figure 4.5: Voltage Latch-based Sense Amplifier

inverter stages M_{n7}–M_{p8} and M_{n8}–M_{p9} and to the bit lines via the two access transistors M_{n5}, M_{n6}. Since inputs and outputs are not isolated from each other, the access transistors are used to disconnect the bit lines from the sense amplifier. This prevents a complete discharge of the bit line transporting the value '0' and saves power and pre-charging time. Figure 4.6 shows the signal diagram of the different SA control signals corresponding to Fig. 4.5. The amplification works in several steps: The sense amplifier inputs are pre-charged through transistors M_{p3} and M_{p5} to equalized the inputs and bias the sense amplifier in the high-gain meta-stable region. At the same time, the bit lines are pre-charged through transistors M_{p6} and M_{p7}. Next, the signals bda_0 and sa_le are set to

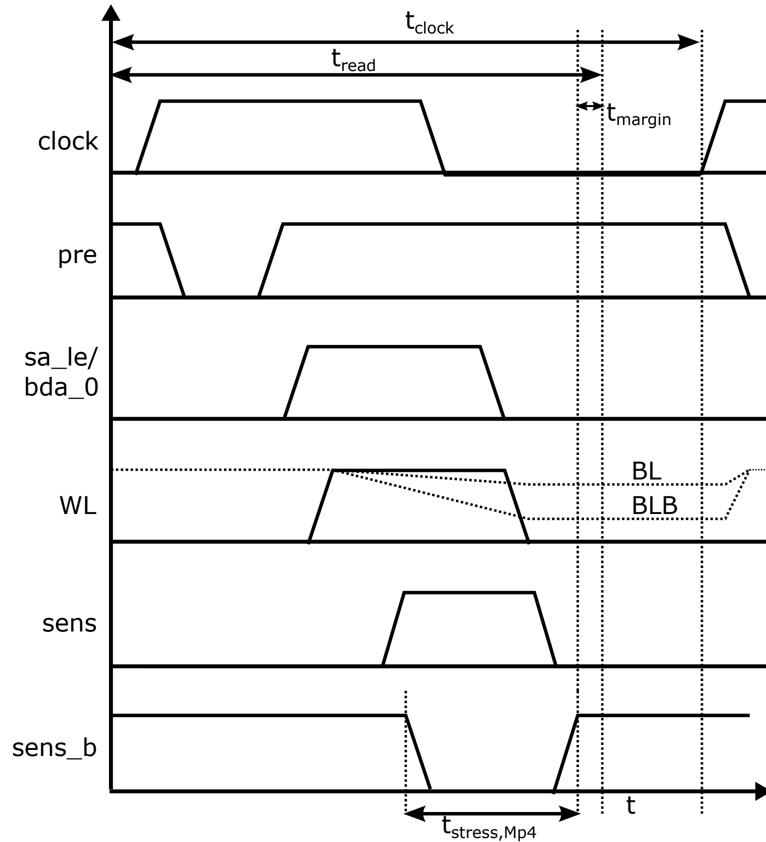

Figure 4.6: Signal diagram to analyze the SA read time and stress times

high. Shortly afterwards, the word line of a specific cell in the SRAM array is enabled and the access transistors M_{n5} and M_{n6} transfer the voltage swings from the bit lines BL and BLB to the internal nodes of the cross-coupled inverters. Meanwhile, the pull-up (M_{p4}) and pull-down network (M_{n3}, M_{4n}) of the inverter pair are still turned off. In the next step, *sens* is set to high and passes through an inverter (not shown here) to set $sens_b$ low, while the access transistors are turned off again to disconnect the bit lines from the internal nodes. Now, the pull-up and pull-down transistors supply the cross-coupled inverters with current. The inverters subsequently amplify the difference between the internal nodes through a positive feedback loop. The outputs become digital outputs *out* and *outb* after passing through the final inverter stage. The time during which the SA is active and performing the described read operation is defined as t_{read}, which is generally shorter than t_{clock} which is the clock frequency of the system. t_{read} includes an additional t_{margin} to account for variations. t_{clock}, t_{read} and t_{margin} will be explained in more detail in section 4.4.2.

4.2 Important SRAM Performance Parameters and Figures of Merit

To perform the operations described above, the SRAM design needs to accomplish a trade-off between important performance parameters like power, speed, reliability, layout area, yield and environmental tolerance. None of these requirements can be optimized without negatively affecting at least one other parameter. This is especially true since design constraints for reading and writing are conflicting challenges as will be explained in detail in Section 4.2.2. Technology scaling to reduce the cell area has a significant impact on the cell robustness due to the dramatically increased sensitivity to process variations and aging. Especially in safety-critical applications it is hence not enough to guarantee correct operation directly after manufacturing but over the complete lifetime of the memory. To optimize the SRAM design for its intended lifetime and application like e.g. high-speed or low-power it is vital to identify the most important figures of merit that measure the sensitivity to noise and transistor parameter variations. In the following, some of the most critical performance metrics are introduced.

4.2.1 Static Noise Margins

Static noise metrics quantify the cell's stability against static noise sources and parameter variations. Especially in the presence of process variations and aging effects, the margins in which a cell can operate and store a bit correctly are becoming increasingly tight. Hence, noise immunity is widely used to quantify SRAM cell reliability.

Static Noise Margin

The static noise margin (SNM) of an SRAM cell is defined as the minimum amount of DC noise required to flip the state of the cell [82]. It characterizes the robustness of a memory cell during hold mode. It can be easily simulated using DC measurements and captures the cell's robustness against transistor parameter fluctuations. Traditionally, it is modeled by applying two voltage noise sources at the internal cell storage nodes, which are swept from 0 to V_{dd}.

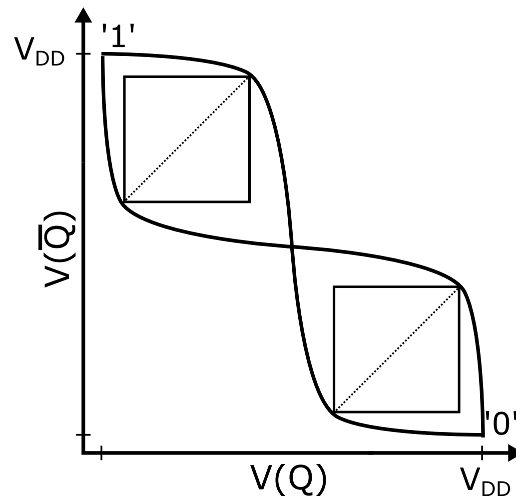

Figure 4.7: Graphical Approach to obtain the SNM using the Butterfly Plot

The lowest voltage level where the cell flips defines the SNM. A graphical representation can be obtained by plotting the mirrored DC characteristics of the cross-coupled inverters, the so-called butterfly curves as shown in Fig. 4.7. The intersection points at the top and the bottom of the curve denote the stable points of the SRAM cell, the crossing in the center the meta-stable point. The SNM can be obtained by measuring the side length of the longest square that can be inscribed into the lobes of the butterfly curve [83]. The SNM value reveals the exact amount that one of the curves can be shifted due to external noise. If the curves intersect only at one point, the cell is mono-stable and has lost its storage capability. Ideal inverters have identical DC characteristics but mismatches within the cross-coupled inverters lead to asymmetric lobes in the butterfly curve. Hence, the SNM is defined as the minimum value of the nested squares in the two lobes of the curve.

Write Noise Margin

The Write Noise Margin (WNM) is also obtained using DC measurements, however this time the DC characteristic of each cross-coupled inverter is measured separately while the word line is driven high and the BL and BLB are set to 0V and V_{dd} respectively. The butterfly curve therefore only has one crossing, indicating that the cell is mono-stable. This allows to write new values into the storage nodes. The WNM is then denoted as the side length of the smallest

square that can be inscribed between the two inverter curves. The WMN measures the extent of mono-stability and hence determines the write robustness of the cell.

Read Noise Margin

The Read Noise Margin (RNM) measures the SNM of an SRAM cell during a read operation and quantifies a bit cell's robustness against read upsets due to DC noise. The measurement is identical to the SNM measurement with the only difference, that the word line is active during the DC sweep of the noise voltage sources.

Since these metrics have been thoroughly investigated in terms of process variations and aging as already pointed out in Chapter 2 this thesis will not further investigate static noise metrics.

4.2.2 Dynamic Noise Margins

Although static noise margins are easy to obtain through DC simulation, they cannot capture the dynamic behavior of memory operations. To obtain an accurate measure of the failure behavior of memories, dynamic noise margins take into account the time dependency of memory operations and aging effects and capture the transient behavior of noise. Dynamic noise margins are commonly expressed as the likelihood that a memory cell is prone to one of the following parametric failure modes [66]. Parametric failures in SRAMs are caused by both time-zero and time-dependent transistor parameter fluctuations [84].

Parametric Failure Modes and Resulting Design Challenges

- **Hold Failures** - *Loss of data due to the inability of a cell to store a given state during the application of a lower supply voltage*

  To reduce the leakage power consumption, most SRAM cell topologies are equipped with a separate power supply (VCC in Fig. 4.4), to supply the SRAM cell in hold mode with a lower supply voltage. The minimum supply voltage for which a cell is able to reliably store a data state is called the Data Retention Voltage (DRV). To reduce power consumption even further,

the peripheral circuitry is also powered down. If the lowering of the cell supply voltage causes the cell content to be destroyed, the cell suffers a hold failure [84].

- **Write Failures** - *Unsuccessful write due to the inability to write a certain value into a cell*

  To initiate a write, the bit line at the internal node storing '1' ($\bar{Q}$ in Fig. 4.4) is pulled to ground. The node is discharged through the access transistor, which is counteracted by the current through the pull-up transistor (L_{p2}). Hence, to flip the inverter the potential of the node $\bar{Q}$ must be pulled below the trip-point of the opposing inverter (L_{n1}–L_{p1}) within the active time of the word line. If the voltage is too high to switch the inverter, a write failure occurs [84]. To overwrite the cell data, the access transistor (pass gate) must be stronger than the pull-up transistor. The pull-up ratio describes the write-ability of the cell: $PR = \frac{(W/L)_{PU}}{(W/L)_{PG}} = \frac{Width/Length_{L_{p2}}}{Width/Length_{L_{n4}}}$ [81].

- **Read Failures** - *Destructive read which flips the stored data in a cell during read access*

  During the read operation, the bit line at the node which is storing '0' (in this example Q) discharges through the path formed by the access transistor and the pull-down transistor. This path forms a voltage divider, which effectively increases the voltage level of this node. To avoid that this so called Read Disturb Voltage (RDV) exceeds the trip point of the opposite inverter ($Ln_2 - Lp_2$), the pull-down transistor must be stronger than the access transistor to reduce the voltage dividing effect. If the RDV exceeds the trip point, the data state stored in the cell is flipped and a read failure occurs [84]. The cell ratio CR quantifies the read-stability of the cell: $CR = \frac{(W/L)_{PD}}{(W/L)_{PG}} = \frac{W_{L_{n1}}/L_{L_{n1}}}{W_{L_{n3}}/L_{L_{n3}}} > 1$ [81]. Higher values of CR increase the read stability and speed but increase the area.

- **Access Failure** - *Wrong data at the output after a read access to a cell due to the inability of the cell to correctly drive the output circuitry (generally the bit lines and sense amplifier) within the chosen access time*

  During a memory read, the accessed cell discharges one of its corresponding bit lines, developing a differential voltage signal. Subsequently, the SA is activated to amplify the differential bit line voltage to a full-swing signal. An access failure occurs, if the small voltage differential cannot be ampli-

fied to a full-swing signal within the specified delay requirement [85]. Access failures therefore happen either because the differential bit line voltage is not high enough to be converted or because the BS cannot be converted by the SA within the given time, e.g. because the operation of the SA is too slow. Hence, every component of the read-path, i.e. the cells in a certain column, the sense amplifier and the signal drivers have their own contribution to the access time. To capture the impact of the individual components of the read-path, the access time can be further broken down into two the two metrics, the Sensing Delay (SD) which reflects SA aging and the Bit line Swing (BS) which reflects SRAM cell aging. The impact of aging on the timing of the SRAM can be characterized by measuring these two parameters which are defined as follows:

Sensing Delay (SD) The sensing delay (SD) is an important timing parameter because it directly correlates with the access speed, i.e. the time delay between requesting a read operation and observing the data value at the output of the memory. We use SD to measure the impact of BTI on the timing of the SRAM. It is an important parameter for quantifying the SA's contribution to the access speed degradation. SD is measured as the duration between the activation of the SA's enable signal and the point in time when the weak differential input voltage has been converted to a full swing output signal (see Fig. 4.8). To capture the SD degradation caused by aging, the following equation will be used:

$$\Delta SD = \frac{SD_{deg} - SD_{nom}}{SD_{nom}} \tag{4.1}$$

Bit line Swing (BS) The bit line swing (BS) is an important parameter for quantifying the SRAM cell's contribution to the access speed degradation and hence the degradation of the sensing delay. During a memory read one of the pre-charged bit lines BL and BLB discharges developing the differential voltage BS. As soon as the SA is enabled, this differential BS is amplified to a full-swing differential output. BS needs to exceed the minimum offset voltage of the SA. When BS is not sufficiently high, the SA may fail to translate the small differential voltage to a full-swing output within the given time. The SA may even latch the wrong output value. BS is measured as the voltage difference between the bit lines BL and BLB

when the SA enable signal is activated (see Fig. 4.9). To capture the BS degradation caused by aging, the following equation will be used:

$$\Delta BS = \frac{BS_{deg} - BS_{nom}}{BS_{nom}} \tag{4.2}$$

The point in time to measure all signal transitions is set to when the respective voltage levels reach 50% of V_{DD}.

Since memories are already the speed bottleneck, access times, which are fast enough to keep up with the processor speed are required. Especially, since the read operation is often considered as one of the most critical operations. Since the access time is the minimum amount of time required to read a bit of data from the memory, usually measured with respect to the initial rising clock edge in the SRAM, SD directly relates to the maximum clock speed the memory is capable to satisfy. It is hence an important parameter to identify SRAM performance degradation under aging, which at the same time captures all components contributing to the degradation of the read-path. BS on the other hand, characterizes the contribution of cell aging to the performance degradation. In this thesis, the two parameters access time - further broken down into SD and BS - and read failures will be used to identify aging in the SRAM read-path.

SRAM designs need to accomplish fundamental trade-offs to reduce overall memory area, while guaranteeing enough functional design margins. Especially since write-ability and read stability pose opposite requirements on the strength of access transistors: For high write-ability, they need to be strong enough to overpower the strength of the pull-up transistors during write (high pull-up ratio). For high read stability, they have to be weak enough to avoid a high

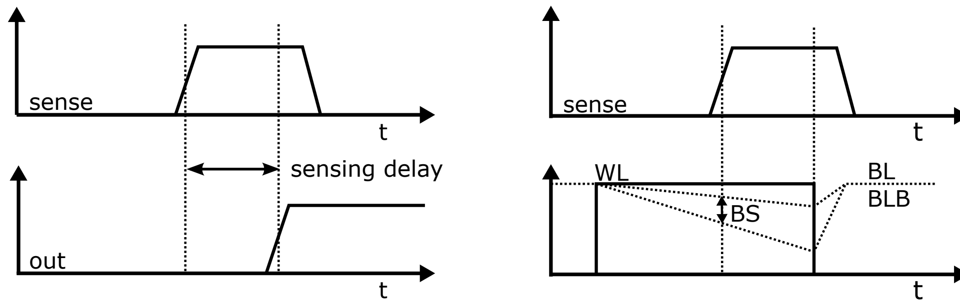

Figure 4.8: Sensing Delay Measurement

Figure 4.9: Bit line Swing Measurement

Read Disturb Voltage, which flips the stored data (high cell ratio). Furthermore, stronger access transistors are required to decrease the access time due to an increased read current for a fast bit line discharge. Accomplishing these trade-offs is inherently aggravated through the rising influence of process variability and wear-out. Transistor parameter variations both at time-zero and during the lifetime of the device disturb the cell/pull-up ratios and symmetry of the cell design [81]. This underlines the necessity of tools which can accurately characterize not only time-zero but also time-dependent variations as early as possible in the design phase.

4.3 BTI Aging of Sense Amplifiers (SAs) and Cells

In this section, we investigate the individual effects of BTI aging on SAs and cells.

4.3.1 The BTI Model

In this work, the aging degradation due to NBTI and PBTI is considered using the long-term model from [67] as described in 3.2.3. Since the simulation setup calculates the threshold voltage shift only once up-front we can afford the high accuracy model. Compared to the overall simulation time of the complete SRAM array on transistor level, the computation time of the accurate aging parameters is negligible. At the same time, this model is capable to accurately capture the recovery effect and hence predict the effect of aging on the SRAM very precisely.

4.3.2 BTI Workload Parameters

The following workload parameters need to be determined to accurately analyze SRAM aging:

Duty Factor (DF) The duty factor $DF_{0/1,i,j,k}$ denotes time $t_{cell,0/1,i,j,k}$, during which an SRAM cell (i, j, k) in row i, column j of a certain bank k stores the value zero/one (denoted as 0/1) as a fraction of the total execution time t_{exe} of an application:

$$DF_{0/1,i,j,k} = \frac{t_{cell,0/1,i,j,k}}{t_{exe}} \tag{4.3}$$

with

$$DF_{1,i,j,k} = 1 - DF_{0,i,j,k} \tag{4.4}$$

Number of Reads from a Cell The number of reads $r_{0/1,i,j,k}$ denotes the number of times a zero/one is read from cell (i, j, k).

Number of Times an SA is under Read Stress For a design which uses L SAs per bank and a word-wise read-out, each word in the core-cell array is multiplexed to these L SAs. Hence, a specific SA l experiences stress whenever a cell in one of the columns $(j * L) + l$ with $j = 0, .., J - 1$ is read. Accounting for the column multiplexing factor, the number of times that an SA l of bank k reads a value zero/one is defined as the sum of the read stress over all columns that this SA is multiplexed to and all cells in these columns:

$$s_{0/1,k,l} = \sum_{m=0}^{I-1} \left(\sum_{n=0}^{J-1} r_{0/1,m,(n*L)+l),k} \right) \tag{4.5}$$

Read Stress Probability (RSP) The read stress probability $RSP_{0/1,k,l}$ for a specific SA l of bank k for reading a zero/one is defined as:

$$RSP_{0/1,k,l} = \frac{s_{0/1,k,l} \cdot t_{read}}{t_{exe}} \tag{4.6}$$

where t_{read} is time during one clock cycle for which an SA l in bank k actually is activated and under stress and t_{exe} is the execution time of the application. The stress of the SA depends on the number of reads it needs to service as well as on the fraction of time it is actively stressed during a read cycle. In Section 4.4.1, we will analyze in detail how to realistically estimate the SA's stress time from the workload profile.

Worst-Case Read Stress Probability ($RSP_{0/1,wc}$) The worst-case read stress probability $RSP_{0/1,wc}$ identifies the SA l of bank k which experiences the highest amount of read stress for reading a zero/one :

$$RSP_{0/1,wc} = \max_{k,l}(RSP_{0/1,k,l}) \tag{4.7}$$

4.3.3 BTI Aging of the Sense Amplifiers (SAs)

The effect of BTI on an SA is strongly dependent on the workload that an application introduces which can be captured as a stress profile for the SRAM array. This stress profile captures RSPs of all SAs, which determines the portion of stress and recovery that each SA in the SRAM experiences. As seen from Eq. (4.6) RSP is defined as the total time of read stress that a certain SA experiences compared to the overall execution time of the application. The stress profile of an application depends on several factors:

Load address profile For an application executed on a typical MCU-type RISC processor, about 10-40% of instructions are loads that require a read access to SRAM data memory. These loads always activate a sub-set of all SAs depending on the load address. Yet, due to the nature of embedded software execution, some memory addresses, and their corresponding SAs, are used very frequently compared to other less often used addresses. Hence, the RSP between SAs can be very unbalanced. For aging, the RSP values of the most read-active SAs are most critical. Here we observed a worst-case RSP of up to 10% for individual SAs as shown in Section 6.1. Previous work [22,78] assumes much larger aging stress values for the SA, which leads to overestimated SA aging.

Load value profile Another factor is whether a '0' or '1' is read from the cell when the SA is used. Hence, we determine RSP seperately for reading '0' or '1'. Due to the symmetry of the SA design as shown in Fig. 4.3, asymmetric aging wear-out has a strong impact: e.g. assuming that the value '0' is constantly read from BL (corresponding to $RSP_0 = 1$) then the transistor M_{n1} experiences always PBTI stress and transistor M_{p2} always NBTI stress. Consequently, their threshold voltage increases due to aging wear-out, leading to a reduced positive feedback, which slows down the read-out process. SD gradually increases until either the access time is violated or until the SA cannot reach a stable operating point anymore such that the read out fails. Fig. 4.10 shows ΔSD (using equation 4.1) for the worst-case read-out condition due to SA aging for different values of $RSP_{0,k,l}$-$RSP_{1,k,l}$ after 3 years at 75°C, nominal V_{DD} and a design with 256 cells in one column in a 32nm technology. Assuming e.g. a worst-case $RSP_{0,k,l}$ of 20% ($RSP_{0/1,k,l}$ cannot be 100% as will be explained in 4.4.2), if the case $RSP_{0,k,l}$-$RSP_{1,k,l}$ = 20% holds, $RSP_{1,k,l}$ = 0% and hence only the value '0' is read from the memory for its entire lifetime, stressing only one side of the SA. Hence,

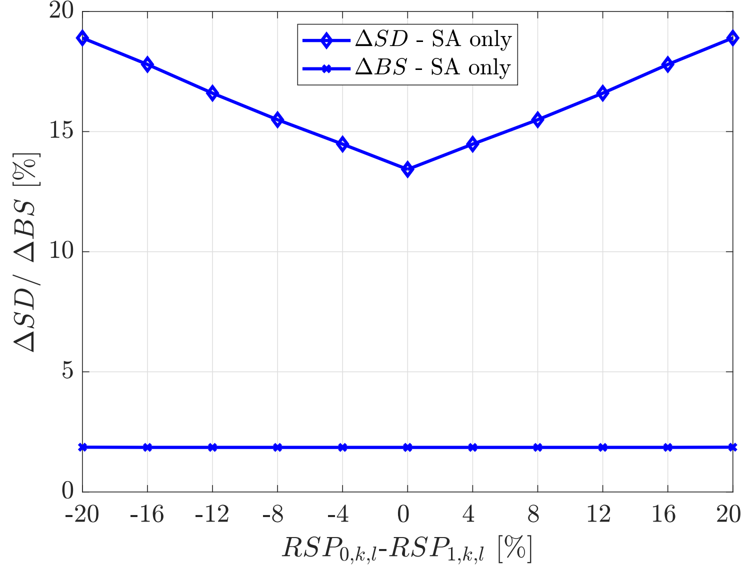

Figure 4.10: Individual Contribution of the SA to ΔSD and ΔBS over $RSP_{0,k,l}$-$RSP_{1,k,l}$. A positive ΔSD / a negative ΔBS degrades the performance of the SRAM.

the worst-case read-out condition results, if yet another '0' is read. If on the other hand $RSP_{0,k,l}$-$RSP_{1,k,l} = -20\%$ ($RSP_{0,k,l} = 0\%$, $RSP_{1,k,l} = 20\%$), only the value '1' is read and the read-out value resulting in the worst-case SD is '1'. If both zeros and ones are read by the SA, the degradation is effectively reduced since the transistors in both cross-coupled inverters are stressed. The figure confirms that ΔSD is lowest for $RSP_{0,k,l}$-$RSP_{1,k,l} = 0\%$ (meaning zeros and ones are read equally often), since the SA is stressed fully symmetrically. This is another evidence that the degradation of SD is strongly workload dependent.

Looking at the curves, there are two factors that can reduce SD degradation: 1) Balancing of the RSP ($RSP_{0,k,l} \approx RSP_{1,k,l}$) of each SA to reach symmetric aging. This will be the minimum degradation for both reading '0' and '1' and, hence, ensures that both cases degrade equally. 2) Balancing of the read accesses across all available SAs to level out $RSP_{0/1,k,l}$ over all banks. Thereby, the $RSP_{0/1,,k,l}$ in particularly stressed banks will be effectively reduced as it is spread over more SAs.

Fig. 4.10 furthermore shows ΔBS (using equation 4.2) caused by SA aging. Interestingly, SA aging has only a relatively small impact on the BS. Furthermore, ΔBS only shows a weak dependency on the workload. SA aging slightly improves ΔBS (resulting in a positive value, since the BS after aging is higher as

compared to its fresh state), which can be explained by the fact that the SA enable signal drivers are also subject to aging. Therefore, the SA activation is delayed, allowing the bit line to discharge deeper which results in a higher BS. Compared to the cell's impact on the BS, however, this effect is much smaller and cannot compensate BS degradation caused by cell aging. It should be noted, that a positive value of ΔBS leads to a performance improvement, since the degraded BS value is higher than its nominal value. In contrast, a positive ΔSD degrades the performance. The results from Fig. 4.10 and Fig. 4.11 were obtained using the application-aware aging analysis flow as introduced in Section 4.5.2 with an array size of ($I = 1$, $J = 1$,$K = 1$).

4.3.4 BTI Aging of 6T SRAM Cells

The effect of BTI on an SRAM cell is strongly dependent on the duty factor $DF_{0/1,i,j,k}$, which determines the portion of stress and recovery that the transistors inside the cell experience. As can be seen from Eq. (4.3), $DF_{0/1,i,j,k}$ is a function of the value that a cell holds and how long the value is stored in that cell. Both depend on the workload. When the value '0' is stored in the cell for its entire lifetime, node Q in Fig.4.3 is constantly zero and $DF_{0,i,j,k} - DF_{1,i,j,k} = 100\%$. Hence, transistor L_{n1} experiences PBTI stress and transistor L_{p2} NBTI stress while transistors L_{p1} and L_{n2} do not experience any BTI stress. Since the threshold voltages of the two stressed transistors increase, the symmetry of the design is destroyed. The positive feedback is reduced, which has been shown to degrade the Static Noise Margin of the cell, making it more susceptible to a bit flip [24]. The same applies to the other pair of transistors (L_{n2} and L_{p1}) if '1' is constantly stored in a cell, resulting in $DF_{0,i,j,k} - DF_{1,i,j,k} = -100\%$. The minimum cell degradation is hence reached if zeros and ones are stored equally often and hence $DF_{0,i,j,k} \approx DF_{1,i,j,k}$. Additionally, cell aging is a major contributor to BS degradation. During a read, one of the pre-charged bit lines BL or BLB discharges through the pull-down transistors. PBTI strongly impacts the pull-down transistors of the core cell and, hence, slows down the discharging, which will decrease BS. A lower BS will, additionally, directly translate into SD degradation. Fig. 4.11 shows ΔBS caused by cell aging for different values of $DF_{0,i,j,k} - DF_{1,i,j,k}$. It can be seen that ΔBS depends heavily on the workload of the cell and leads to an increase in ΔSD. The relative contribution of the cell to ΔSD depends on the strength of the the pull-down transistors in the core cell and hence their sizing [22]. The design used in this work is a low-density cell. It

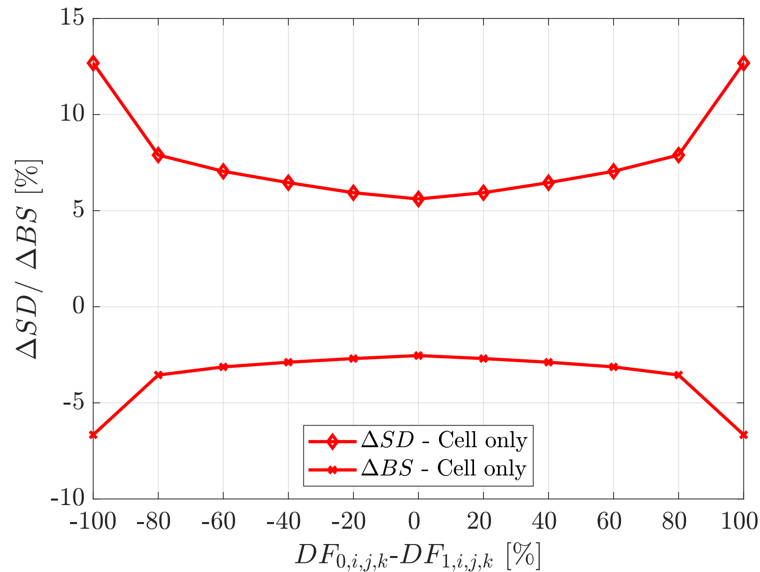

Figure 4.11: Individual Contribution of the SA to ΔSD and ΔBS over $DF_{0,i,j,k}$-$DF_{1,i,j,k}$. A positive ΔSD / a negative ΔBS degrades the performance of the SRAM.

has almost equally strong pull-up and pull-down transistors. Hence, the contribution of the cell to ΔSD is quite strong when the cell is exposed to high aging stress. The cell aging needs to be taken into consideration jointly with SA aging when investigating the read-path degradation. Otherwise, SD degradation is clearly underestimated.

4.4 Application-aware Analysis of BTI Aging (AppAwareAge)

To avoid pessimistic guardbands based on worst-case assumptions without compromising memory reliability, aging prediction requires realistic workload information. Furthermore, an accurate prediction of the aging behavior caused by BTI requires to take into account all sub-components.

As was pointed out in the last section, both SA and cell aging and, hence, SD and BS depend strongly on the workload. In this section we therefore introduce a novel reliability tool for SRAM Design-for-Reliability to accurately predict the aging degradation considering workloads from embedded application executed

on an industrial MCU by simulating the complete SRAM read-path based on [86] in a two-step process. The proposed method is applied to analyze the contributions of SA and SRAM cell aging for various workloads, temperatures and supply voltages on an industrial SRAM Design. Our work focuses on the impact of aging on SD and BS, but it is possible to modify the flow to also handle other figures-of-merit like SA offset voltage drift. The simulation framework consists of two main parts as shown in Fig. 4.12:

1. a high-level simulation step based on an Instruction Set Simulator (ISS) to profile the stress of the cells and SAs using memory traces obtained during execution of the application, and
2. a low-level aging analysis of the complete SRAM array based on an aging-aware netlist generation and low-level aging simulation step using the stress profiles that have been generated in the first step.

Both steps are detailed in the following.

4.4.1 Stress Profiling by High-level Simulation

Memory Tracing

The stress profile for the SRAM is generated from a memory trace of the application workload, which is executed on the MCU. The trace contains the addresses and corresponding values, which are written to and read from the SRAM with their corresponding time stamps. The ETISS-ML from [87] is used to execute the application workloads for typical input sets to generate the memory traces, but any other simulation that is able to track memory accesses can be deployed as well. An ISS provides a higher performance compared to RTL simulation allowing to run a larger number of application workloads to obtain more memory trace data.

Generation of Cell Stress Maps

The aging stress of an SRAM cell in the memory is characterized through $DF_{0/1,i,j,k}$. $DF_{0/1,i,j,k}$ is extracted for each cell from the memory trace by analyzing access addresses, values and their corresponding time stamps. The difference between two time stamps of two consecutive writes to the same address will result in the

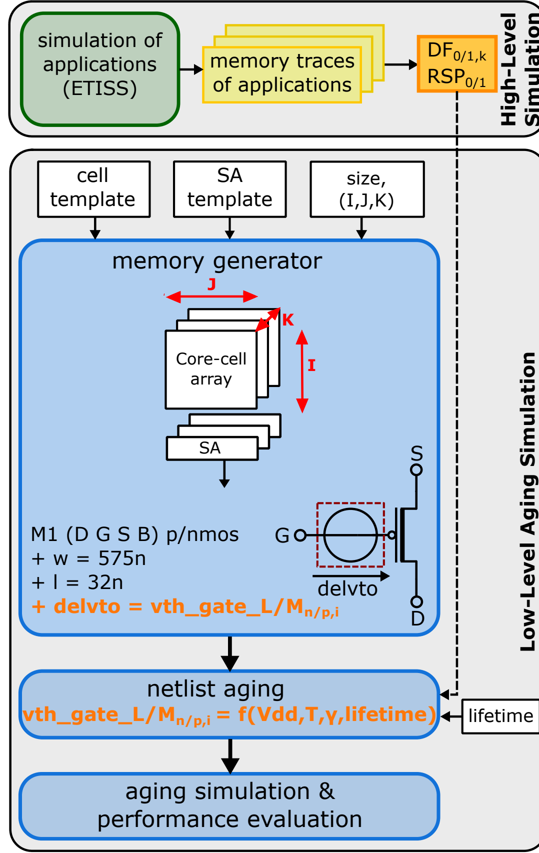

Figure 4.12: Application-Aware Aging Analysis Flow

time that a value was stored. Thereby, we analyze each cell individually. Hence, it is possible to gather the exact time that each individual cell stored the value zero or one during the execution of an application. With this information we create the cell stress maps $DF_{0/1,k}$ containing the stress profile of all SRAM cells in one bank k as:

$$DF_{0/1,k} = \begin{pmatrix} DF_{0/1,0,0,k} & \cdots & DF_{0/1,0,(J*L)-1,k} \\ \vdots & \ddots & \vdots \\ DF_{0/1,I-1,0,k} & \cdots & DF_{0/1,I,(J*L)-1,k} \end{pmatrix}$$

In the above matrix, J is the number of words, $(J * L)$ the number of bits in a row for a word-size of L, I is the number of rows and K is the number of banks. The matrix entry $DF_{0/1,i,j,k}$ represents the duty factor of cell j in row i of a certain bank k for storing either '0' or '1' as defined in Eq. (4.3).

Generation of SA Stress Maps

The aging stress of an SA in the SRAM array is characterized through $RSP_{0/1,k,l}$. $RSP_{0/1,k,l}$ is extracted for each SA from the memory trace by analyzing the number of reads for each SA as described in 4.3.2.

To obtain the stress profile of all SAs in an SRAM for a specific application, the RSP values $RSP_{0/1,k,l}$ of each SA are arranged in a SA stress map $RSP_{0/1}$ with K rows and L columns where K is the number of banks and L is the word length. The overall stress map for reading '0' or '1' is then given as

$$RSP_{0/1} = \begin{pmatrix} RSP_{0/1,0,0} & \cdots & RSP_{0/1,0,L-1} \\ \vdots & \ddots & \vdots \\ RSP_{0/1,K-1,0} & \cdots & RSP_{0/1,K-1,L-1} \end{pmatrix} \tag{4.8}$$

4.4.2 Aging Analysis by low-level Transistor Simulation

Definition of SA read time t_{read}

The SA is not active during the complete clock cycle, in which a read access is performed. There is a certain read time $t_{read} < t_{clock}$, for which the SA actually is activated and under stress. Figure 4.6 shows the signal diagram of the different SA control signals corresponding to Fig. 4.3. The SA operation starts with the pre-charging of the SA inputs via signal *pre* to equalize the inputs and ends with turning off the power-up and power-down network of the SA via signals *sense* and $sense_b$. For the time t_{read} between pre-charging and turning off the SA, the SA is considered as active. t_{read} should account for the SA activation time at nominal conditions plus an additional t_{margin}, which is added to the nominal t_{read} to consider PVTA variations in order to avoid timing errors.

Little information can be found on the corresponding t_{read} but in [88] it is reported to be around $t_{read} = 1ns$. For example for a processor running on 500MHz $t_{clock} = 2ns$. Hence, even if the same SA would be used for reading from the SRAM in every clock cycle, the RSP of that SA would only be around 50%, as the SA is active in half of the read clock cycle.

Definition of Transistor Duty factor $\alpha_{M_{n/p,i}}$ in the SA

In addition to the acknowledgment that a SA is not active during the complete clock cycle, it is important to realize that most transistors are not under stress for the complete t_{read} as can be seen in Fig. 4.6. Hence, it is important to consider the transistor-level duty factor $\alpha_{L/M_{n/p,i}} = \frac{t_{stress,L/M_{n/p,i}}}{t_{read}}$ representing the fraction of time a transistor in the circuit actually experiences BTI stress during read-out.

Definition of Transistor Duty factor $\alpha_{L_{n/p,i}}$ in the Cell

Of course duty factors in the cells also vary for different transistors. However, determining the duty factors in the cell is much simpler, since only one transistor pair $L_{n,i}$ - $L_{p,i}$ experiences 100% of the BTI stress (see Fig. 4.3).

The duty factor of each transistor is determined once through simulation up front.

Low-level Aging Analysis of the Complete SRAM Array

The low level aging analysis of the complete SRAM array consists of three steps:

1. aging-aware memory netlist generator that creates a netlist for a given configuration of rows, columns and banks while adding aging parameters for each transistor,
2. setting of aging parameters in the generated netlists according to the identified SA and cell stress profiles and
3. simulation of the aged netlist and SRAM performance evaluation.

The cell and SA stress maps $DF_{0/1,k}$ and $RSP_{0/1}$ are the inputs to the aging-aware memory netlist generator. Additional inputs to the memory netlist generator are a template netlist of a 6T SRAM cell, a template netlist of the SA shown in Fig. 4.3, the lifetime in years, the desired memory size in kB as well as the size in terms of number of rows, columns and banks. The memory netlist generator first creates the netlist of the SRAM array according to the model in Fig. 4.2 and adds a variable drift parameter $v_{th_}gate_L/M_{n/p,i}$ to the instance parameter $delvto$ of each transistor to incorporate the threshold voltage shift.

In the second step the threshold voltage shifts for NBTI and PBTI are calculated from the DF and RSP stress maps, the supply voltage, the die temperature as well as the desired memory lifetime for each transistor. For SA aging, the specific V_{th}-shift that each transistor experiences is a function of the RSP and results to $v_{th}_gate_M_{n/p,i} = f(V_{dd}, T, \gamma, lifetime)$ with $\gamma = RSP_{0/1,k,l} \cdot \alpha_{M_{n/p,i}}$ where $\alpha_{M_{n/p,i}}$ is the duty factor of transistor $M_{n/p,i}$ and $gate$ being the name of the gate signal. For cell aging, the V_{th}-shift of a transistor is a function of the DF and results to $v_{th}_gate_L_{n/p,i} = f(V_{dd}, T, \gamma, lifetime)$ with $\gamma = DF_{0/1,i,j,k} \cdot \alpha_{L_{n/p,i}}$ where $\alpha_{L_{n/p,i}}$ is the duty factor of transistor $L_{n/p,i}$. Finally, the netlist is "aged" by setting the variable drift parameter to a fixed value according to the threshold voltage shift as predicted by equation (5.8).

The third step sets up the simulation of the SRAM array by automatically generating all necessary input stimuli and measurements for SD and BS depending on the configuration and finally starts the Spectre simulations. The simulation measurements are then evaluated by determining ΔSD and ΔBS for each read-path and logging failed read-outs.

4.4.3 Speeding-up the Analysis

Since the proposed method analyzes the complete SRAM array, excessive simulations are needed to evaluate the read-out degradation of every word in the memory. Owing to the fact that the generated stress maps already contain lots of information about the aging behavior of the SRAM array, we can exploit this knowledge to considerably reduce the number of simulations and, hence, speed-up the analysis.

Based on the information of the cell and SA stress maps DF and RSP only the read-path showing the highest stress values for SA and cell needs to be measured, which restricts the simulations to one read-out of only one word in one bank.

4.5 Aging Mitigation in SRAMs

4.5.1 Mitigation of AGIng Circuitry (MAGIC)

In this section, we introduce a mitigation circuit based on [89] that can counter the aging degradation of SAs. First, we discuss the basic on-chip SRAM Design-for-Reliability concept, before giving the implementation details.

Basic Mitigation Concept

While different application generally show very different workloads and hence stress profiles for the SRAM, we observed with our workload-aware aging analysis that the stress profile is often very unbalanced across the memory's address range. For example, in SRAM-based on-chip data memories, frequently accessed addresses correspond to often used data inside the data section or to the stack of the executed program. The stack and data section are often at fixed locations as accesses to on-chip data memories are done without a memory management unit. Other memory addresses, which are located between stack and the static data section are often used less frequently or not at all as many programs do not make full use of the available memory or some variables are rarely read. This behavior is not preferable in terms of SA aging: the frequently accessed address ranges may only decode to just a few banks of the SRAM such that the SAs of these banks experience exacerbated stress resulting in fast aging.

As a solution, MAGIC mitigates aging by regularly modifying the mapping between memory addresses and cells in the physical memory. The stress is leveled more evenly over all physical memory locations, which is known as wear-leveling. Specific SAs can be spared from exaggerated wear-out, if wear-leveling is implemented in such a way that cells from different banks are used for frequently accessed data. Yet, such a scheme must come with small overheads in terms of area and performance. MAGIC achieves this by implementing the re-mapping with a single XOR-gate array as will be discussed in the following.

XOR-based Address Remapping

For the address remapping, an extremely simple and efficient remapping circuit is used that consists only of XOR gates. Fig. 4.13 shows the circuit for an exemplary 4-bit input. Each XOR gate has as input one address and one offset bit.

Remapping of the address is achieved because the bitwise XOR operation results in a modified physical address given the received physical address and an offset value provided by a 4-bit counter. Every time the offset value increases, the original physical address is mapped to a different modified physical address. The behavior of the address remapping is presented in Table 4.1, which illustrates the behavior of the output bits s_i in dependency of the address input bits d_i and the counter value c_i with $i = 0, 1, 2, 3$.

This XOR-based remapping circuit is superior to barrel shifters, which were proposed in the state-of-the-art [29]. The XOR circuit is more compact imposing small area overhead, is fast with only one single additional gate delay on the address path and also remaps addresses that are all zeros or ones. In contrast, rotation shift operations on all zero/one values do not modify those values.

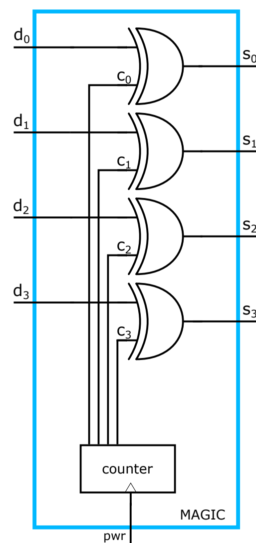

Figure 4.13: Circuit diagram for MAGIC XOR Remapping Block

SRAM Architecture using MAGIC

As discussed in 4.1.1, Fig. 4.2 shows an overview of the SRAM architecture. In the standard configuration, a bank decoder directly receives the bank address from the address bus, decodes the value and selects the bank to access. To mitigate aging in the SAs of the SRAM array, the bank address is first passed through the MAGIC block. MAGIC remaps the bank addresses received by the

Table 4.1: Modified bank addresses from XOR remapping circuit

Modified bank address $s_0\ s_1\ s_2\ s_3$		Offset value (from counter) $c_0\ c_1\ c_2\ c_3$				
		0000	0001	···	1110	1111
Original bank address $d_0\ d_1\ d_2\ d_3$	0000	0000	0001	···	1110	1111
	0001	0001	0000	···	1111	1110
	···	···	···	···	···	···
	1110	1110	1111	···	0000	0001
	1111	1111	1110	···	0001	0000

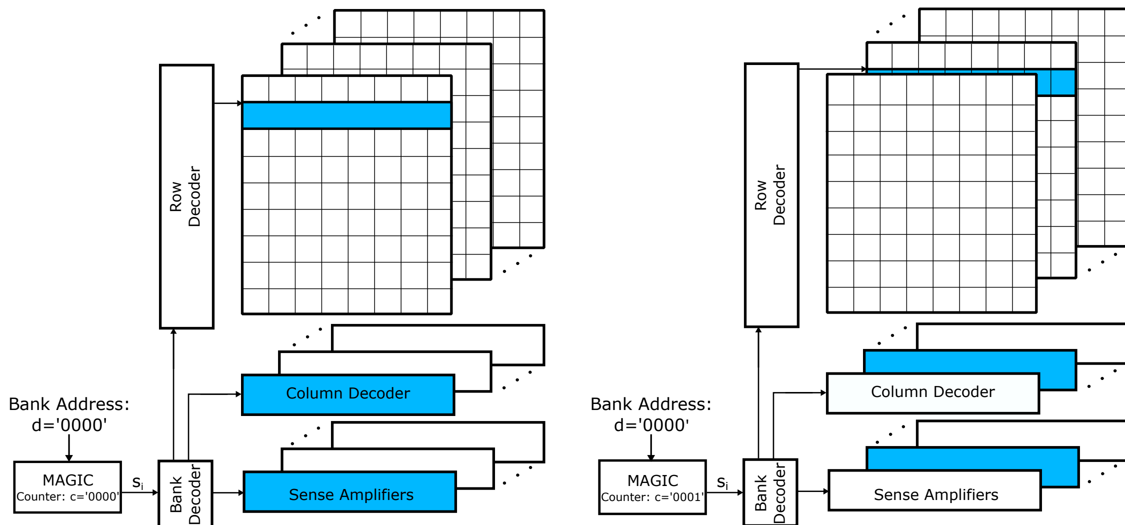

Figure 4.14: Bank Remapping Principle with MAGIC

address interface to a modified bank address. This modified bank address is forwarded to the bank decoder.

Fig. 4.14 demonstrates the modification of the physical bank address by MAGIC. The offset value to the XOR-gate array that results in the modification is provided as a counter value. Assuming e.g. a bank address d = '0000', the first bank is selected if the counter value is zero. If the counter value increases to one, the second bank will be selected instead and so on, following the remapping scheme of Table 4.1. This bank remapping based on XOR modifications achieves a high aging mitigation in the SAs as the wear-out is leveled evenly across all banks.

Since the bank address modification is applied for both read and write accesses, the proposed method does not result in data corruption. Yet, it must be ensured that the memory holds no live values when the counter value is updated. For

this, MAGIC requires a safe update method to change the counter value, e.g., the remapping is only applied at selected operating points, where no live data is stored in the memory (e.g. power up, reset). A realization for this would be to use a counter with a non-volatile memory (e.g. FLASH) or a volatile memory that has its own power supply and is incremented every time a circuit is powered up or resetted. This can be done unlimited times as the counter may overflow to return to the first address mapping. As alternative, also schemes with random values as offsets would work.

MAGIC can be also moved in front of the row or column decoder stage or even in front of more than one decoder depending on the desired wear-leveling. An insertion of this block in front of the row and word decoder stage should enhance aging mitigation in the cells. An integration in front of the word decoder is only possible, if it can be guaranteed that the memory is only addressable word-wise, since otherwise the stored data is corrupted due to shuffling least and most significant bytes.

The focus of our work was to mitigate aging in the sense amplifiers since we believe aging degradation in the sense amplifiers is more critical than aging mitigation in the cells. Obviously, cells, which rarely/never accessed experience a high amount of stress, however a failure in the sense amplifier is especially critical since a failure of the sense amplifier destroys the read-out of a whole column, while cell aging only renders a single bit useless.

Lifetime Prediction

A new lifetime prediction is proposed to investigate the impact of MAGIC on the lifetime of the memory. We define the end-of-life of the device by assuming that at design time a worst-case spec of $125°C$ at $-10\%V_{dd}$ has to be fulfilled which results in a worst-case sensing delay SD_{wc}. SD_{deg} defines the degraded SD after aging. If now SD_{nom} is the sensing delay at nominal temperature ($25°C$) and nominal V_{dd}, the deviation of the worst-case from the nominal case $SD_{wc} - SD_{nom}$ constitutes the available margin for the SRAM array. Any delay deviation larger than that will result in an access time violation. Hence we define the end-of-life of the device with the following equation:

$$SD_{deg} - SD_{nom} > f_{safety} \cdot (SD_{wc} - SD_{nom}) \tag{4.9}$$

For safety-critical applications an additional factor f_{safety} might be beneficial to add some extra safety to the margin. Since our work unfortunately neglects all kind of process variations and mismatch (mostly due to simulation time restrictions and since they were heavily investigated in many other works) which play a significant role in the performance evaluation of SRAMs, we account for these variations by adding margin to the worst-case degradation in order to reduce pessimism. Assuming the end-of-life of a circuit at $125°C$ at $-10\%V_{dd}$ while not taking into account variations seems to be an unrealistic condition, especially for SRAMs which knowingly suffer significantly from process variations. f_{safety} adds timing margin to the measured sensing delays and hence accounts for process variations in the harsh end-of-life conditions. We regard f_{safety} as a tuning parameter to take into account that the SRAM should still be functional at $125°C$ at $-10\%V_{dd}$ and additional variations. Since this factor is only used for post-processing of the simulation results, it can easily be tuned to a value appropriate to the investigated design and operating conditions without repeating any simulations.

4.5.2 Aging-Aware SRAM Design Exploration (SDE)

Especially in safety-critical SoCs a precise estimation of the application specific aging behavior is crucial to predict the memory lifetime and add adequate guardbands. For the reliability of these systems it can be beneficial to analyze the aging behavior of different memory architectures for the intended application before deciding on a specific design. Array granularity (e.g. number of banks/rows/words) has a significant impact on the aging behavior of the memory, since the decoding of addresses is dependent on the chosen array granularity. Hence, the same address results in a different physical location in the memory array depending on the chosen granularity configuration. This can be used as a benefit, since appearing workloads can be distributed onto more banks if a granularity with a higher number of banks is chosen. Similarly, the sensitivity to aging can be reduced by attaching fewer cells onto one bit line. Hence, for an intended set of applications some of these architectures are more reliable than others. An exploration of all possible memory configurations in terms of aging early in the design phase can significantly improve memory reliability, avoid wasteful design margins and guarantee reliable behavior of the memory for its intended lifetime. SDE can therefore be used to find the optimal trade-off between area, power and reliability.

In this chapter the second aging mitigation scheme called SRAM design exploration framework (SDE) [86] is presented. It incorporates the application-aware aging analysis for on-chip SRAMs AppAwareAge introduced in Chapter 4.4. To analyze aging, the tool generates and characterizes memories of different array granularity with detailed simulations to find the most reliable configuration in terms of aging for the intended set of applications. Again, the sensing delay SD is examined and cell and SA aging is considered to identify the degradation of the complete read-path. It is however possible to trivially modify the flow to handle other figures-of-merit like SA voltage drift.

Aging Behavior Characterization of Memory Arrays with different Granularity

Fig. 4.2 shows that the granularity of the SRAM core array has three degrees of freedom, the number of rows I, columns J and banks K. In the following, the aging behavior of the SRAM core array is explored by adjusting the array granularity using these three discrete parameters I, J, K. It was pointed out before that although different applications generally show very different workloads, certain address ranges are accessed more frequently than others. These ranges correspond to often used data inside the data section or to the stack of the executed program. To explain the impact that each of the parameters has on the memory reliability the first observation is that these heavily accessed addresses usually decode to only a few banks. Hence, the SAs of these banks experience exacerbated stress and hence higher aging rates. Since the decoding of the addresses is dependent on the array granularity, an increase in the number of banks can spread the workload across more banks and hence mitigate aging. The penalty for a larger number of banks however is, that the area of the memory is increased since each bank needs its own L SAs for a word-size of L. Furthermore, decreasing the number of rows decreases the bit line capacity, which can help to mitigate aging as well, since the bit line swing is higher for fewer rows as less parasitics are present. Adjusting the number of words however, should not have any impact on the aging behavior and is only necessary to retain the correct memory size.

Aging-Aware SRAM Design Exploration Framework

The aging-aware SRAM design exploration framework utilizes the application-aware aging analysis for on-chip SRAMs introduced in Chapter 4.4. The flowchart

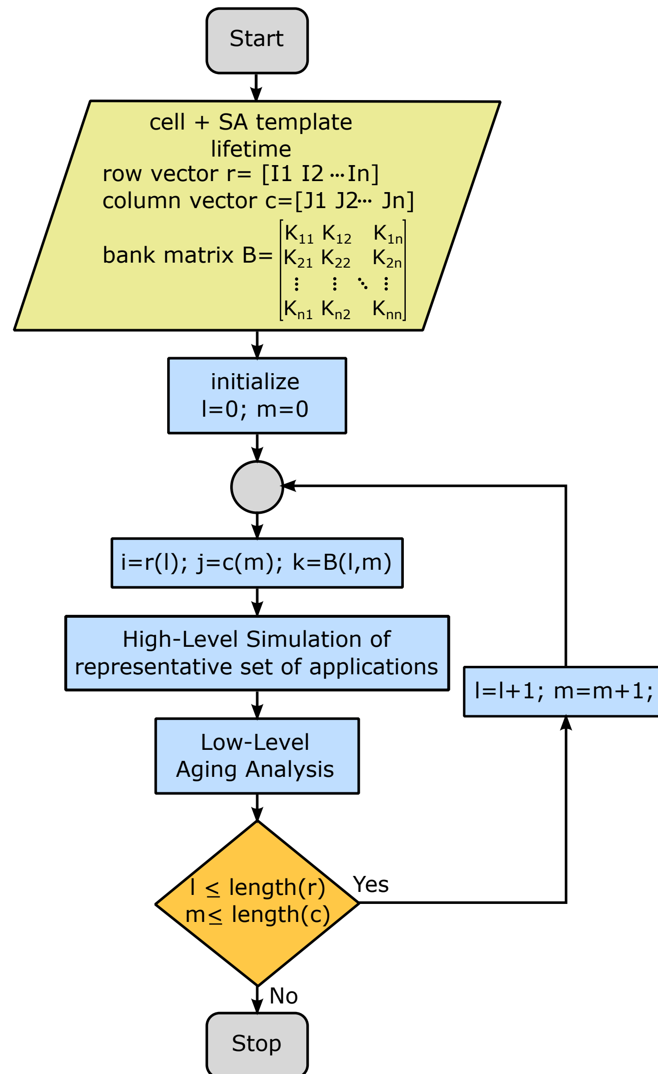

Figure 4.15: SRAM Design Exploration Framework Flowchart

of SDE is shown in Fig. 4.15. SDE takes as inputs a template netlist of a 6T SRAM cell and a template netlist of the SA and the lifetime in years. Furthermore, all possible memory configurations with respect to the memory size must be given in terms of a row vector r which contains the desired number of rows, a column vector which contains the number of columns and bank matrix that captures the possible number of banks for each configuration. SDE starts from initializing two control variables l and m which gradually increase their size and assign the size of the memory in terms of number of rows, columns and banks to the variables I, J, and K. Each combination of I, J, and K therefore must result in the

desired memory size. As next step, the high-level simulation step as described in Chapter 4.4 is carried out to profile the stress of the cells and SAs using the memory trace obtained during execution. Each array configuration possesses its own stress maps DF and RSP since the distribution of the workload depends on the address decoding and hence the granularity. Subsequently, the low-level aging simulation step including an aging-aware netlist generation is executed. Finally, l and m are increased by one and the variable assignment of I, J, and K as well as the high- and low-level simulation step are repeated in a loop fashion until $l > length(r)$ and $m > length(c)$.

4.6 Summary

This chapter introduces a novel reliability tool for SRAM Design-for-Reliability incorporating a new workload-aware aging analysis for on-chip SRAMs that incorporates the workload of of embedded applications executed on the MCU while considering aging in the complete read-path and its control signals. According to this workload, we predict the performance degradation in the memory and its end of lifetime.

We furthermore propose MAGIC, an aging mitigation circuitry that levels read stress evenly across the complete SRAM array, hence effectively mitigating the wear-out of the SAs. The proposed mitigation scheme can significantly improve the degradation of the SRAM, which is extremely important for safety-critical systems to guarantee full functionality till the end of life of the device.

Moreover, the proposed tool can be used as an SRAM design exploration framework (SDE) that generates and analyzes memories of different array granularity to find the most reliable configuration in terms of aging for the intended set of applications.

5 Runtime Power, Temperature and Aging Monitoring on FPGA Prototypes

Although the transition to multi-core made it possible to keep up with the performance demand and Moore's law, it is not going to solve the dark silicon problem and the rising impact of chip temperatures, which result in increasing reliability issues in the long term. Fig. 5.1 shows the amount of dark silicon usable logic on a chip for different technology nodes as per projections of [12], showing that designers face up to 90% of dark silicon when high operating frequencies are applied, meaning that only 10% of the chip's hardware resources are useful at any given time. Consequently, the increasing amount of dark silicon directly reflects on the chip performance to a point where multi- and many-core scaling provides zero gain [90].

Additionally, multimillion transistor system-on-a-chip architectures face a dramatically increasing rate of temporary and permanent faults. Processing capabilities for different cores, even of the same type, can differ significantly depending on process variations and time-dependent fluctuations like temperature, aging and supply voltage. Static and central management concepts to control the execution of all resources might soon reach their limits for SoCs with a large number of cores [13].

A way of dealing with dark silicon issues and variations across cores is the introduction of resource-aware computing concepts, which can adjust the amount of allocated resources (e.g. cores) for a running application according to its temporal needs in a self-organizing manner. For large systems with thousands of cores on a chip, resource-aware programming is vital to obtain high utilization as well as computational, energy and power efficiency [91]. Heterogeneous systems thereby seem to be a promising option since they provide more flexibility as they contain highly customized processing units that can run specific tasks and applications at high performance and low energy [92].

Consequently, in future dark silicon chips, the usage of on-chip resources needs to be power, temperature and aging-aware. A way to achieve this is hardware

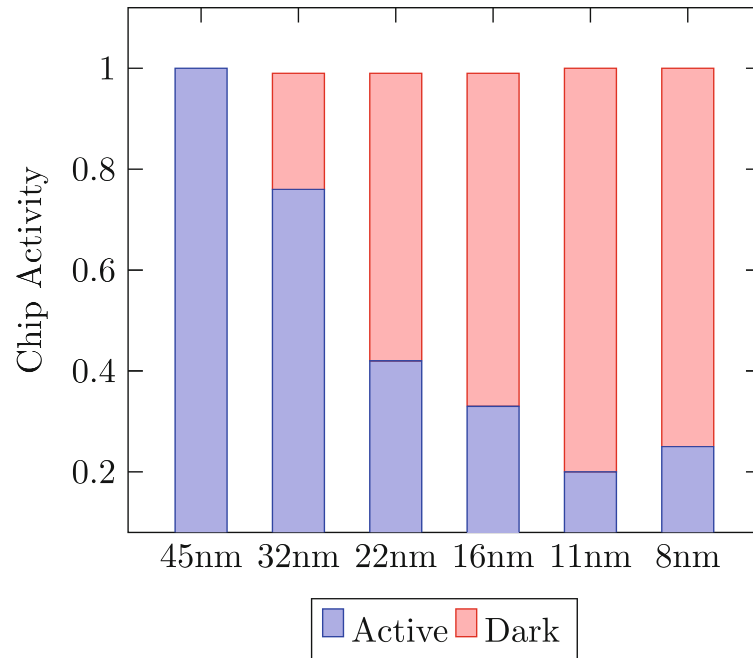

Figure 5.1: Projections of Dark Silicon in future Technology Nodes [90]

monitoring, which provides crucial information about the current hardware health status. Knowledge about the current physical hardware properties like power consumption, temperature and degradation can be exploited to locate the best suitable processors to execute an application. Power monitoring data can efficiently be utilized for load balancing across cores. Temperature monitors can avoid arising hotspots and the resulting unreliable behavior and exacerbated wear-out. Monitoring of the aging status makes it possible to detect and avoid critical reliability conditions by re-allocation to more reliable resources for the requesting application. This provides also chances to exploit aging recovery effects, since heavily degraded cores can be left idle until their secure functionality is re-established. FPGA prototyping can be utilized prior to the implementation of the ASIC to develop and evaluate runtime management and resource allocation strategies early during the design phase of multi-core processors.

In this work a real-time power, temperature and aging monitor system (eTAP-Mon) for FPGA prototypes of MPSoCs is proposed. The FPGA monitor system emulates data of the target ASIC design through dedicated models inside the hardware monitors, which have been developed and characterized based on data acquired from the target ASIC design. The emulation approach models the be-

havior of ASIC power monitors based on an instruction-level energy model and supports dynamic voltage-frequency-scaling while considering also the power behavior of both data and instruction caches. The temperature monitors are based on a linear regression model obtained from detailed thermal offline simulations with the thermal model HotSpot [93]. The aging monitors are based on a critical path model. They model the dependency of the threshold voltage (V_{th}) shift on temperature, supply voltage and age and compute the decreasing timing margin due to aging. An accelerated aging emulation is possible to predict aged ASIC behavior. Hence, this FPGA emulation enables the early evaluation of runtime management strategies. Results were obtained from the implementation of the monitor system on an FPGA board. The monitor system is evaluated for a selected operating scenario for nominal process corners, where it provides useful insights into the power, temperature and aging behavior of the system.

The remainder of this Chapter is organized as follows. First, a novel resource-aware programming concept is introduced in Section 5.1, that can utilize the proposed monitor system to develop management strategies which counter arising reliability threats and future dark silicon issues. In Section 5.2 the concept of FPGA prototyping and its benefits are shortly explained. Afterwards, the monitoring approaches for the emulation of power, temperature and aging and their implementation in hardware are described in detail in Sections 5.3.1, 5.3.2 and 5.3.3, respectively. Finally, the Chapter is concluded in Section 5.4.

5.1 Invasive Computing

A novel paradigm for designing and programming future parallel computing systems is called "invasive computing". Figure 5.2 shows a typical heterogeneous invasive multi-processor architecture including several loosely-coupled processors (standard RISC CPUs and specialized invasive cores called i-Cores) as well as tightly-coupled processor arrays (TCPAs). Invasive computing suggests a resource-aware programming support where programs get the capability to request and temporarily claim processor, communication and memory resources of neighboring processors. Using the claimed resources, portions of code of high parallelism degree are executed in parallel. Once the program terminates, or if the degree of parallelism is lower again, the program may de-allocate these claimed resources again [13]. Invasive computing consists of the three distinct phases as depicted in Figure 5.3: invade, infect and retreat. In the invade phase,

an initial claim constituting computing resources (cores), communication (e.g. bandwith) and memory (e.g. caches) is issued. The infect phase starts the execution of the application on the allocated resources requested in the claim. When the execution completed, the claim size can either be increased by issuing a re-invade, or decreased by proceeding to the retreat phase. At this stage it is also possible to re-infect the claimed resources with another application. After the program is terminated, the retreat phase deallocates the claimed resources and frees them for other applications.

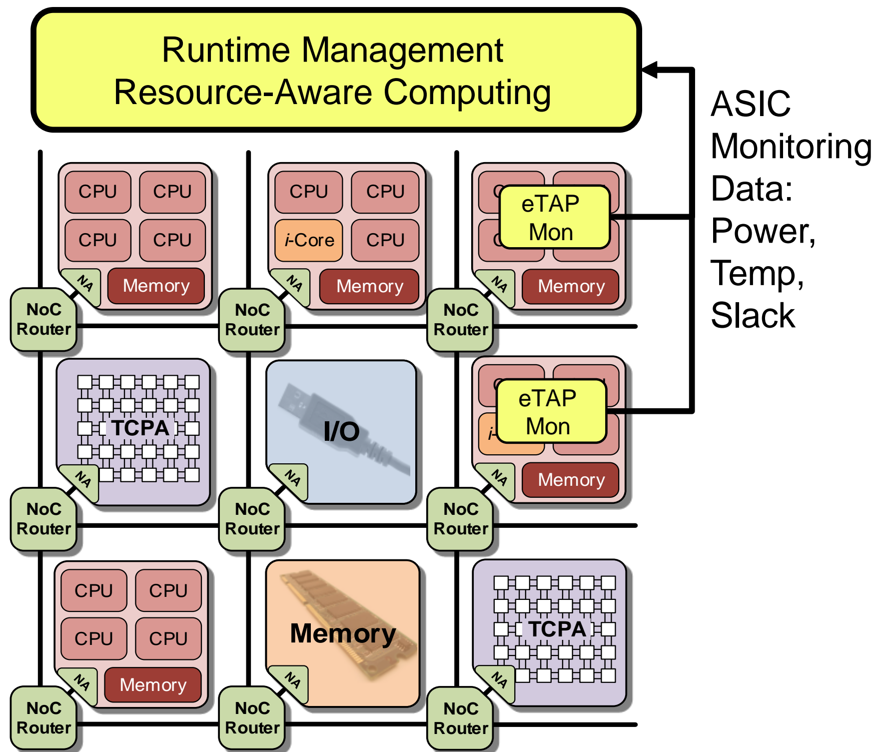

Figure 5.2: Generic invasive multi-processor architecture including several loosely-coupled processors (standard RISC CPUs and invasive cores, so-called i-Cores) as well as tightly-coupled processor arrays (TCPAs)

A major advantage of this programing paradigm results from the fact, that resources are only claimed if they are available and actually needed. Hence, resource utilization is dramatically increased compared to statically mapped applications. Furthermore, the degree of parallelism of an application can be adapted with regards to the available resources. In the case of heterogeneous systems, the programmer is given the freedom to execute specifically tailored implementa-

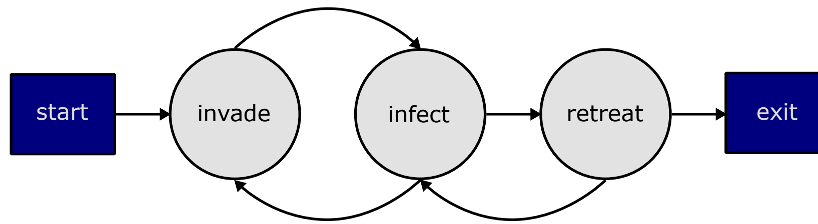

Figure 5.3: State Chart of an Invasive Program

tions of an algorithm on highly customized processing units to achieve a higher computing performance or lower power consumption. Since the resource allocation is organized in a decentralized manner, it is highly scalable which is an important property for large multi-core systems [91]. Hence, this heterogeneity can help to circumvent the problem of dark silicon.

Monitoring data provides vital information about the current hardware health status like current power consumption, temperature and aging. It can therefore be exploited in resource-aware computing strategies to avoid the utilization of unreliable hardware components or retreat from resources which are becoming to hot or consume too much power. Furthermore, management strategies can utilize the monitoring information in advance to mitigate hotspots and exacerbated aging.

However, for the development of such resource-aware computing strategies, monitoring data for power, temperature and aging from the actual ASIC implementation if the multi-core system is required which usually does not exist during the design phase. To support the early investigation of runtime management strategies in resource-aware computing, this work proposes to place hardware monitors on the FPGA Prototype of the ASIC instead. Therefore, the developed monitors (eTAPMon) are placed on top of the CPUs as shown in Fig. 5.2 to deliver vital information about power, temperature and aging to the runtime management system.

5.2 FPGA Prototyping in Multi-Core Processors

The utilization of cycle-accurate software simulators to explore hardware-software configurations is not feasible any more in modern embedded multi-core architectures. Instead, the focus recently shifted towards field programmable gate array

(FPGA)-based hardware emulation. FPGA prototypes provide a reconfigurable, highly parallel and inexpensive platform to emulate parallel architectures at hardware speed allowing to run real benchmark programs to obtain realistic core activities. With FPGAs, a far more rapid turnaround for new hardware is possible compared to the traditional hardware development cycle, suitable to effectively tackle the evolving problems regarding parallel processing. Therefore, FPGA platforms can support the software community to take advantage of the potential of parallel microprocessors, by providing a platform that enables the early development of software and advances the cooperation with the hardware community.

Hence, to evaluate and optimize resource-aware computing strategies during the design phase, FPGA prototyping is a suitable method to obtain a functioning hardware platform of the target processor long before its silicon implementation is available. Apart from the benefit of a higher performance of prototyping approaches compared to simulation-based approaches, the RTL of the MPSoC can be used to generate the FPGA prototype, thus limiting the overhead compared to high-level system models, that need additional modeling effort. The combination of power, speed, flexibility, observability and reproducibility makes FPGAs the ideal platform for future multi-core system developments.

To acquire the necessary data for the development of resource allocation and runtime management strategies, hardware monitors can be placed on the prototyping platform as well. However, the evolution of power, temperature and aging on the FPGA is significantly different from the targeted ASIC. In order to get monitoring data from the prototype which corresponds to the ASIC data, the behavior of the ASIC must be emulated through models inside the hardware monitors. The models must be established through characterization of the target ASIC design to correctly reflect the ASIC. Using these models, the emulated monitor system can deliver useful ASIC data for the evaluation and development of management and resource allocation strategies as will be shown in the following.

5.3 ASIC Monitoring System

In this section we present the proposed emulated power, temperature and aging monitor system (eTAPMon) for FPGA prototypes [94]. Fig. 5.4 shows an

overview of the eTAPMon including the monitored FPGA prototype of a quad-core compute tile of an MPSoC. The monitor system consists of three blocks: The power monitor that monitors CPU signals to compute power values (described in Section 5.3.1), the temperature monitor delivering steady-state temperatures based on these power values (see Section 5.3.2) and the aging monitor that converts an increasing V_{th}-shift to a decreasing timing margin (see Section 5.3.3). Power, temperature and aging can be emulated for the corresponding ASIC system via dedicated models inside the hardware monitors, which have been developed and characterized based on data from the target ASIC design. The eTAPMon delivers data characterized from an ASIC LEON3 processor with a TSMC 40nm technology running at a maximal clock frequency of 1200MHz. The suggested methodology can easily be adapted for other processor architectures.

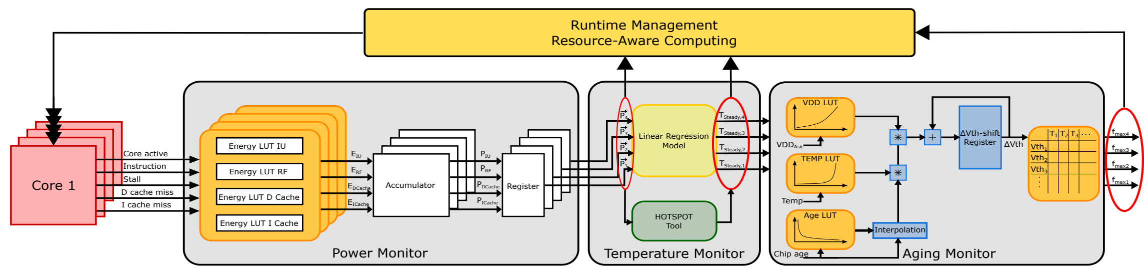

Figure 5.4: Implementation of the eTAPMon for a LEON3 quad core design.

5.3.1 Emulation of the Power Monitor

The first stage of the emulated monitor system is the power monitor generating energy values based on CPU signals through the evaluation of an instruction-level energy model. There is one power monitor per core. The monitor taps the following signals from its core for our example design (see Fig. 5.4): Core active, the 32-bit instruction bus, stall as well as the data cache and instruction cache miss signals. The power monitor, which is shown in more detail in Fig. 5.5, translates these signals to energy values based on instruction-energy Look-Up-Tables (LUTs). The LUTs store the energy value that each component consumes per clock cycle for different instruction types. For each core, the energy values of four blocks are evaluated: Pipeline (IU), register file (RF) and both cache energies (D Cache, I Cache). The LUT entries are generated via offline RTL simulations

of a single LEON3 Core in TSMC technology and consist of the dynamic as well as the static energy, denoted as $E_{dyni,j}$ and $E_{stati,j}$ respectively, where $i = 1;2;3;4$ are the indices of the cores and $j = 1;2;3;4$ are the indices of the considered blocks (IU, RF, D Cache, I Cache). While static energy is always consumed by each block, dynamic energy is only dissipated if the core active signal is high. Furthermore, register file is inactive when the stall signal is high. For the two caches, additionally the cache miss signals must be high in order for the blocks to consume dynamic energy for fetching data from 2nd level memory.

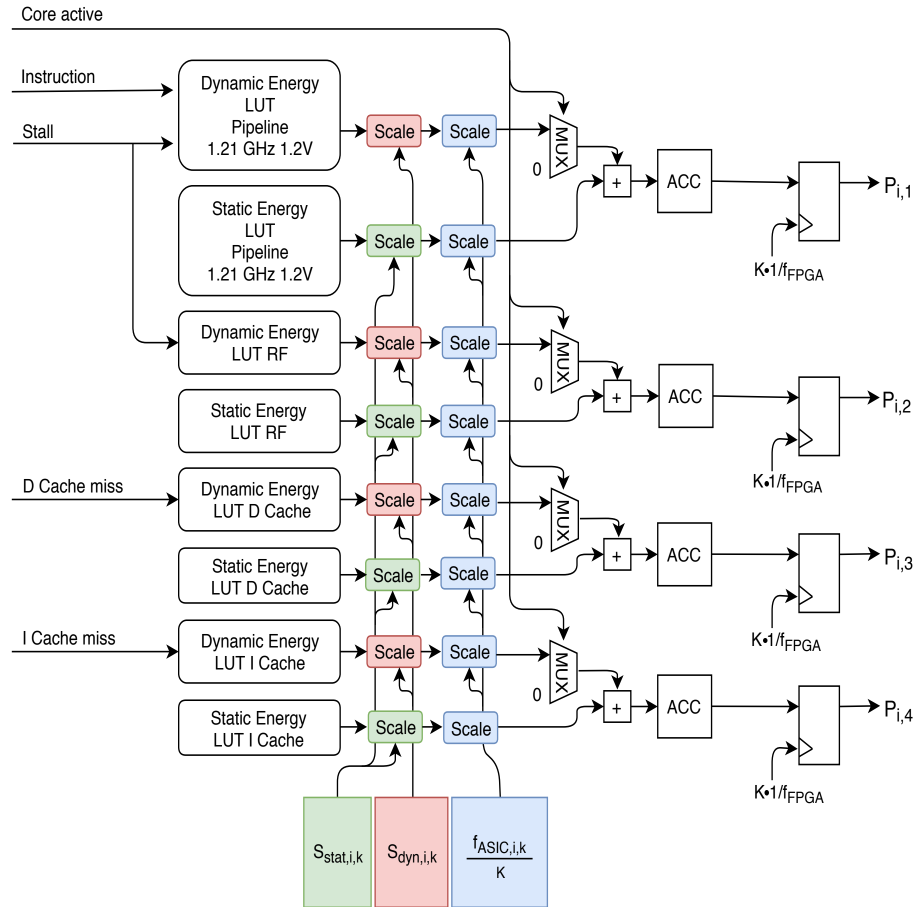

Figure 5.5: Emulated Power Monitor.

Dynamic Voltage-Frequency-Scaling

The energy values that are stored in the LUTs are only specified for one reference ASIC supply voltage $V_{DD_{ref}}$. The values are scaled when core i runs at a different

voltage $V_{DD,i,k}$ in cycle k. The static energy depends linearly on the supply voltage whereas the dynamic energy shows a quadratic dependency:

$$s_{stat,i,k} = \frac{V_{DD,i,k}}{V_{DD_{ref}}}, \quad s_{dyn,i,k} = \left(\frac{V_{DD,i,k}}{V_{DD_{ref}}}\right)^2$$

The energy values are internally accumulated over K=100,000 clock cycles in the accumulator. To emulate an ASIC design, the difference between the ASIC core frequency and the FPGA frequency needs to be considered. The ASIC would accumulate the same energy in faster time as it runs at higher frequency $f_{ASIC,i,k}$. Therefore, we compute the average core power $P_{i,j}$ by dividing the accumulated and scaled energy by the time interval that would pass for ASIC computation $\Delta t_{i,k} = K/f_{ASIC,i,k}$:

$$P_{i,j} = \sum_{k=1}^{K} (s_{stat,i,k} E_{stat,i,j,k} + s_{dyn,i,k} E_{dyn,i,j,k}) \cdot \frac{f_{ASIC,i,k}}{K}$$

The resulting monitored power is written to the output register of the power monitor and forwarded to the temperature monitor. The runtime management calculates the required scaling factors and sets the corresponding scaling registers in each monitor to mimic the voltage-frequency scaling of the ASIC on the FPGA prototype.

5.3.2 Emulation of the Temperature Monitor

The modeling approach for the real-time emulation of an ASIC temperature monitor as shown in Fig. 5.6 uses the resulting power values $P_{i,j}$ to calculate the corresponding steady-state core temperatures T_c of a core c for a certain monitoring period. As mentioned earlier, the temperature of a core not only depends on its own power consumption, but also on the activity and power consumption of the neighboring cores. This is called neighbor effect. To account for that, the chosen model fits not only the core's own power values, but also the power values of the neighboring cores. The temperature-power relation is close to linear. Hence, a linear regression model is used:

$$T_c = \sum_{i=1}^{4} \sum_{j=1}^{4} \left(a_{i,j,c} \cdot P_{i,j}\right) + a_{off,c}$$

where $a_{i,j,c}$ are the corresponding model coefficients that need to be fitted (16 coefficients per each core in total) and $a_{off,c}$ is the respective offset for each core. For the fitting of the model, a real test set of power values is tracked from the FPGA (as shown in Fig. 5.4) while the cores execute a given workload. The power values from the FPGA trace are then used in detailed offline simulations with the thermal model tool HotSpot [93]. For the 16 power inputs the tool delivers four steady-state temperatures for the four evaluated blocks (IU, RF, D Cache, I Cache) for each core. In addition to the power consumption obtained from our power monitors, a processor floorplan (die layer) of the architecture (2x2 LEON3 multi-core design) is an input for the HotSpot simulations. We

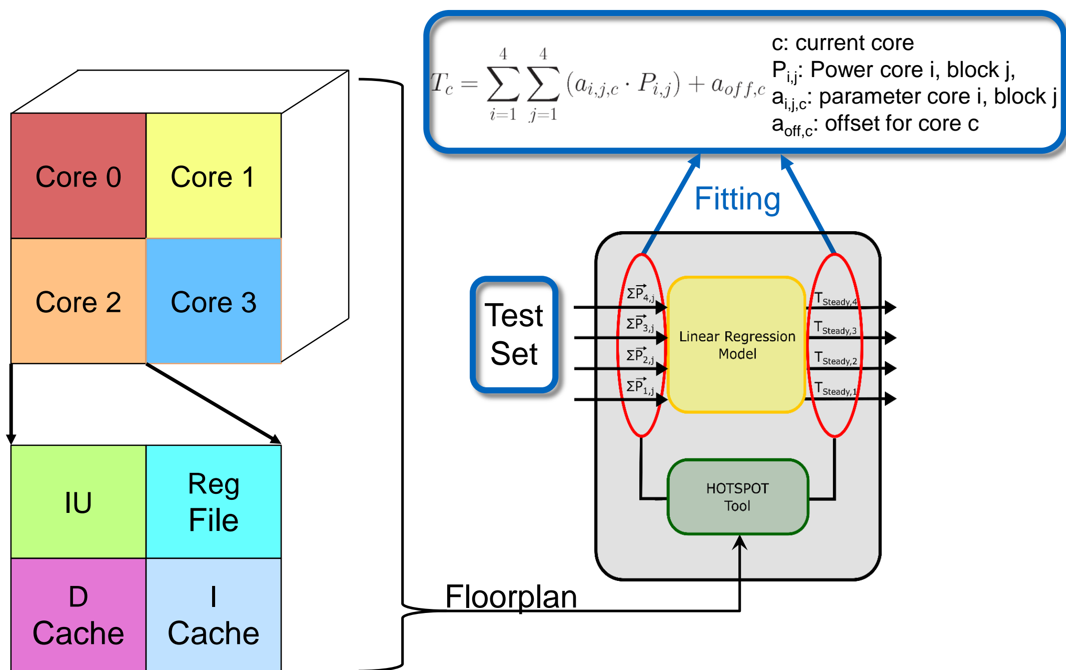

Figure 5.6: Emulated Temperature Monitor.

assumed the monitoring period is large enough to reach steady-state temperatures. Since the monitoring information should be conservative, the maximum temperature from each part of the core is chosen to serve as an upper bound [50]. After fitting the model, the obtained regression equation is finally implemented in hardware and has to be solved once per core to deliver the maximum temperature for the current power values. As the monitoring period is 100k cycles and thus no fast computation is required, a shared data path with one multiplier and adder is used to limit resource usage.

5.3.3 Emulation of the Aging Monitor

Aging monitors observe the decreasing timing slack on selected critical paths of the system. The slack is defined as the difference between the required arrival time and the actual arrival time of a signal. As mentioned earlier, NBTI has been identified as the crucial mechanism threatening device reliability and will hence be the focus of our aging monitoring approach.

The BTI Model

NBTI increases the threshold voltage of pMOS transistors and hence degrades the drain current causing a temporal performance degradation which results in a growing delay of logic gates.

In this work, the aging degradation due to NBTI is considered using the low accuracy model from S. Nassif (personal communication, November 10, 2016) as described in 3.2.3. Since the aging monitor needs to calculate the threshold voltage shift for every clock cycle, the low accuracy model provides a reasonable trade-off between hardware overhead in the FPGA and monitoring accuracy.

Implementation of the BTI Model in the Aging Monitor

To model the aging behavior for MPSoC prototypes, first the threshold voltage shift that an individual transistor experiences, based on the given supply voltage and core temperature, is calculated with the following equation:

$$\Delta V_{th}(chipage) = 0.05 \cdot exp(\frac{-1500}{T_c}) \cdot {V_{DD}}^4 \cdot chipage^{\frac{1}{6}} \cdot \alpha^{\frac{1}{6}}$$

where T_c equals the current core temperature in K obtained from the temperature monitor, V_{DD} the current core voltage, *chipage* the lifespan (age) of the chip in seconds and α the duty cycle. As tracking the duty cycle of individual transistors would cause a large area overhead on the FPGA, we set α to 0.5, resulting in a conservative approximation of aging since the sub-function $\alpha^{1/6}$ already reaches almost 90% of its final value as can be seen in Fig. 5.7. Note, that with this equation it is not possible to calculate aging with changing temperature, V_{DD} and duty cycle parameters. Therefore, the time is discretized

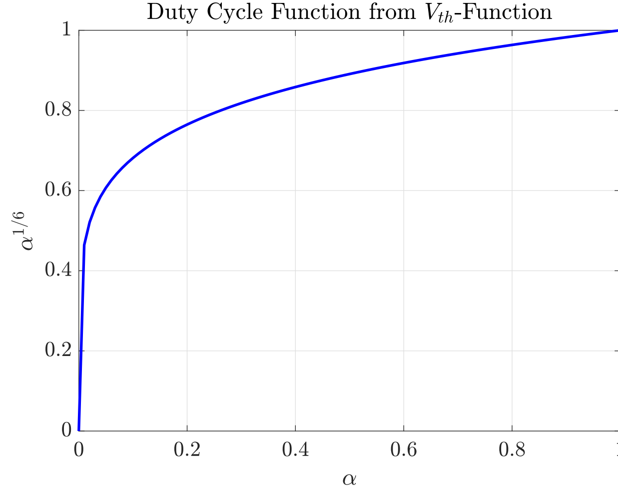

Figure 5.7: Behavior of Duty Cycle in Aging Function.

with $m \cdot \Delta chipage$, where for each time step the core temperature T and V_{DD} are assumed to be constant.

$$\Delta V_{th}(m+1) = \Delta V_{th}(m) + \frac{\partial \Delta V_{th}(chipage)}{\partial chipage} \cdot \Delta chipage$$

The resulting equation is then used to model the dependency of the V_{th}-shift on varying use profiles of V_{DD} and T over the increasing age of the chip. To utilize a function like this in an FPGA, three specific lookup tables were created and implemented in hardware to approximate the non linear sub-functions ${V_{DD}}^4$, $\exp(1500/T)$ and $\frac{\partial \Delta V_{th}(chipage)}{\partial chipage} = chipage^{-5/6}$ (see Fig. 5.4: VDD LUT, Temp LUT, Age LUT).

To translate the threshold voltage shift caused by NBTI to an increasing gate delay and hence a reducing slack, a critical path model, characterized from the target ASIC LEON3 design, is suggested and explained in the following.
The critical path model is created in three steps as shown in Fig. 5.9. In the first step, aged netlists of all available gates in the target library are created. To achieve this, the original netlists of the gates available in the standard cell library are prepared with a constant voltage source of a predefined ΔV_{th}-value at each gate terminal to incorporate the pMOS threshold voltage shift.

Since the aged netlists are generated only for discrete values of ΔV_{th}, a logarithmically spaced sampling vector $\Delta V_{th,l}$ is chosen to sample l values from $chipage$ since the slope of the threshold voltage shift changes strongly close to

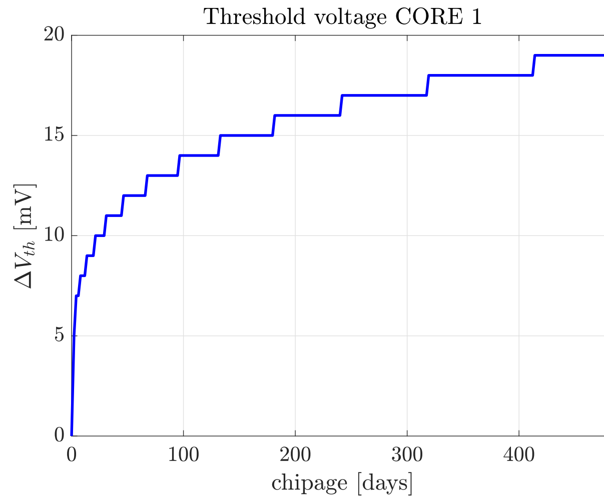

Figure 5.8: ΔV_{th}-values monitored from the FPGA Prototype during the Standard Emulation Scenario.

$chipage = 0$ and flattens with growing values of time. This can be seen in Fig. 5.8 which shows monitored ΔV_{th}-values obtained from the FPGA prototype for the exemplary emulation scenario described in Section 6.3. Hence, a logarithmically spaced *chipage* sampling vector allows for a linear interpolation between the sampling points.

For the second step, a commercially available library characterizer (SiliconSmart [95]) has been used to create aged libraries for each individual $\Delta V_{th,l}$-sample. The standard characterization flow can be applied with the simple difference that the tool is provided with aged gate netlists. In addition to the aged netlists, a reference library file from the TSMC library as well as a transistor model is needed as input to the re-characterization flow. In our study we used a standard Predictive Technology Model (PTM) [96]. The output of the library characterization tool is an aged library file (.lib) that can be converted to a technology file (.db) and used for a subsequent Static Timing Analysis (STA). Since the speed of the circuit is not only influenced by an increasing threshold voltage shift but also by the temperature, we add a second dimension to our characterization and additionally incorporate different operating temperatures to fully exploit the spectrum of our use profiles. For the temperature sampling a linear spaced sampling vector and linear interpolation are applied to sample the values T_l. As re-characterization has to be done only once up-front, the method scales very well with the library size.
In the third step, a critical path analysis of the target design is conducted with

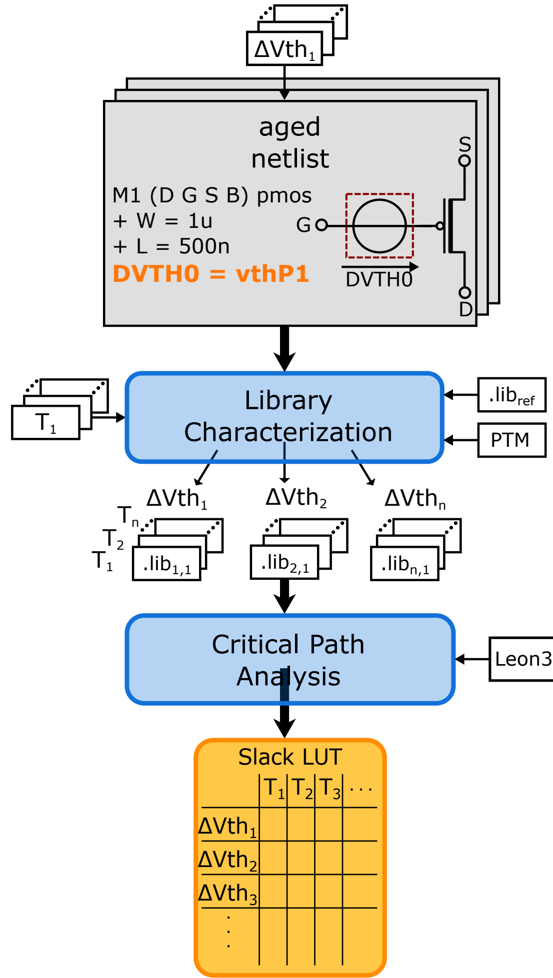

Figure 5.9: Generation of critical path model for the emulated Aging Monitor.

each of the generated aged libraries and a Verilog description of the circuit. Again, we use a commercially available tool (PrimeTime [97]) while following the standard STA flow to determine the slack of the critical path. The slack and hence the available frequency margin for the current age can be determined and captured in a 2-dimensional Slack-LUT containing the slack of the critical path for each sampled $\Delta V_{th,l}$ and T_l. For our study we use the Instruction Unit (IU) of the LEON3 processor, assuming the aging monitor is attached to the critical path of the IU.

Aging Acceleration

Usually, no aging is observed when executing a test program on the FPGA prototype as only very small shifts in ΔV_{th} are recognized. In order to be able to predict aging for a longer lifetime of the system, it is possible to trigger the aging monitor more often. For the emulation this means that the threshold voltage shift and hence the slack are not calculated for the real-time values of V_{DD} and T every $\Delta chipage$ period anymore. Instead, within the same period, ΔV_{th} and slack are evaluated multiple times and accumulate already within one $\Delta chipage$ period as shown in Fig. 5.10. This artificially extends the runtime for each core when looking at the aging behavior. Although the parameters V_{DD} and T are not changing in real-time any more, the accelerated aging results in a comparable amount of stress as running the test program multiple times consecutively. Since we don't consider any recovery cycles in the program and work with a fixed duty cycle, the accelerated aging feature thus predicts the degradation with good accuracy under the assumption that the cores see a fixed workload over their complete lifetime.

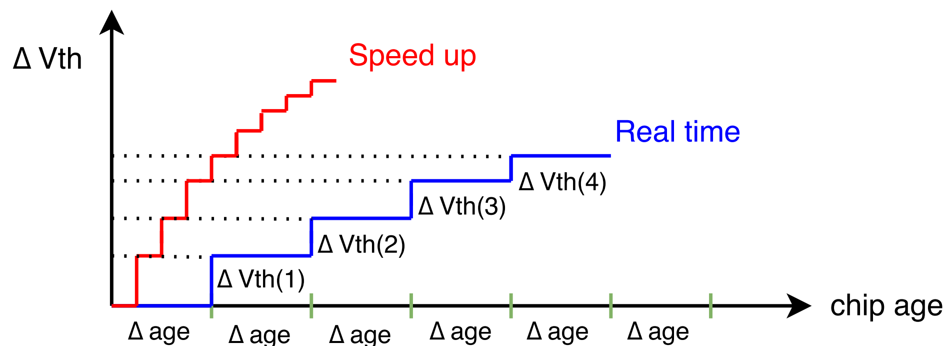

Figure 5.10: Accelerated aging

5.4 Summary

In this chapter a real-time power, temperature and aging monitor system (eTAPMon) for FPGA prototypes of MPSoCs is proposed that is able to predict reliability threats and can supply this information to the runtime management to develop efficient power management and resource allocation strategies.

6 Experimental Results

6.1 Application-Aware Aging Analysis (AppAwareAge) and Mitigation of AGIng Circuitry (MAGIC)

In this section, results are shown for the novel reliability tool AppAwareAge and the efficiency of the MAGIC mitigation. First, we apply the proposed method to analyze the contributions of SA and SRAM cell aging for various application workloads, temperatures and supply voltages. Then, we compare the aging of the read-path with and without using MAGIC as mitigation scheme. The proposed lifetime prediction is utilized to investigate the impact of MAGIC on the lifetime of the memory.

6.1.1 Experimental Setup

We apply the presented application-aware aging analysis on a 128 kByte on-chip data memory of an OpenRisc 1000 processor (OR1k) [98] which runs at a frequency of 750MHz. The time the SA is active during one clock cycle is assumed as $t_{read} = 1ns$. The SRAM array is arranged in 16 banks with 256x32x8 arrays (256 rows, 32-bit word width, 8 column multiplexing factor). For the MAGIC circuit, we assume that the counter value increases after each application run and that the same application is running for the considered time of aging. Memory traces of six applications including an encryption algorithm (AES), sorting algorithms (Heap, Isort), image processing and compression (Edge, JDCT) and a digital filter algorithm (IIR) are considered as application workloads to obtain the stress maps DF and RSP of the SRAM array. For the low-level transistor simulation a 32nm Predictive Technology Model [96] is used.

The following experiments were conducted to illustrate the workload-dependent aging analysis, analyze the individual contributions of cell and SA aging on the read-path degradation and highlight the effectiveness of aging mitigation using MAGIC:

Memory Utilization and Application Stress Profiles For the chosen set of applications, the memory utilization, the stress maps of the cells (DF) and the stress maps of the SAs (RSP) of the read-path, which experiences the worst-case aging stress in the memory configuration, are analyzed.

Application-Dependent BTI Impact We investigate the two cases 'SA aging only' and 'combined cell and SA aging' of the before mentioned worst-case read-path to show the distinct contributions of SA and cell aging to the SD degradation (ΔSD) and BS degradation (ΔBS) for 3 years of aging at 75°C and nominal supply voltage ($0.9V$) in Section 6.1.3. The results of the 'combined cell and SA aging' case are compared to the worst-case DFs and RSPs after the application of MAGIC in Section 6.1.4. Furthermore the effect of MAGIC on ΔSD and ΔBS is shown.

BTI Impact over Time ΔSD and ΔBS of the worst-case read-path is investigated over time for the identified best-case and worst-case workloads for the two cases 'SA aging only' and 'combined cell and SA aging' in Section 6.1.3 to illustrate the aging trend over time. Furthermore, the time-dependent aging of the 'combined cell and SA aging' case with and without the proposed MAGIC mitigation technique is compared in Section 6.1.4.

BTI Impact over Temperature According to the identified worst-case and best-case workloads, ΔSD and ΔBS of the the worst-case read-path is investigated in Section 6.1.3 for three different temperatures (25°C, 75°C, 125°C) for both cases 'SA aging only' and 'combined cell and SA aging' and three years of aging at nominal supply voltage. The results of the 'combined cell and SA aging' case with and without the proposed MAGIC mitigation technique are compared in Section 6.1.4. This illustrates the temperature dependency.

BTI Impact over supply Voltage For the worst-case and best-case workloads, the impact of the supply voltage for three different supply voltages ($-10\% V_{dd}$, nominal V_{dd}, +10%V_{dd}V) on ΔSD and ΔBS of the worst-case read-path is analyzed at nominal temperature (25°C) for both cases 'SA aging only' and 'combined cell and SA aging' in Section 6.1.3. Again the combined aging case without MAGIC is compared to the results using MAGIC mitigation in Section 6.1.4.

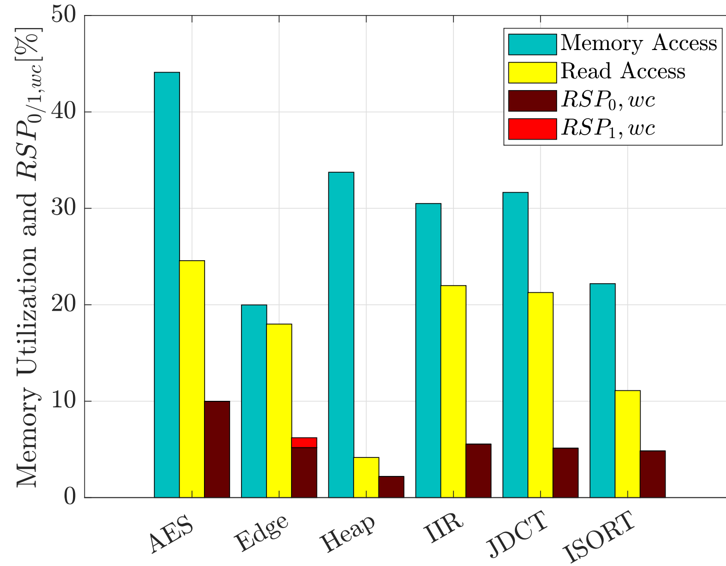

Figure 6.1: Memory Usage and Worst-case RSP (over all SAs) for different applications

Lifetime prediction Using the proposed lifetime prediction, the expected lifetime of the SRAM is predicted for 'combined cell and SA aging' with and without mitigation for all workloads and a harsh use case of 90°C and -5% V_{dd} for the combined aging case.

6.1.2 Memory Utilization and Application Stress Profiles

Fig. 6.1 shows the memory usage per application. 'Memory access' displays the percentage of time that each application actually accesses the memory compared to the overall execution time. 'Read access' shows the time that each application spends reading the memory. The read accesses between the different applications differ significantly between the workloads. This can be explained by the fact that some applications are performing more internal computations, while others are using more load and store instructions which ultimately lead to read and write accesses to the memory. $RSP_{0/1}$ reveals the workload of the SA of the read-path which is experiencing the highest amount of aging stress (worst-case) of the SRAM memory for different applications. The memory column of this read-path would potentially be the first one to fail completely (ignoring process variability). For most applications the value '0' is read significantly more often in the worst-case SA than '1', which clearly results in asymmetric aging of this SA.

Since each word has 32 bits, it is more likely that many of the most significant bits of these words are '0' for the majority of the execution time, if the application is making computations with small positive values. This can also be observed in the extracted memory traces for each application. However, the workload of the SAs is generally much lower than often assumed. The previously reported RSP values were significantly overestimated in [22, 78]. One reason is that the percentage of read instructions seems generally overestimated (up to 80% read instructions per application) since the work in [22] uses artificial workloads. Another reason is that the work in [78] uses workloads for L1 data and instruction caches as discussed in 2 which generally show very different patterns. In contrast, this work uses realistic workloads from embedded applications. A third reason is that it is not clear whether it is considered that the SAs are not active for the complete clock cycle of the read. The worst-case cell stress is not shown here since $DF_0 \approx 1$ for all applications. This can be explained by the fact that only a very small range of addresses is used by the applications and hence most cells store the value '0' for their entire lifetime.

6.1.3 Analysis of the individual Contributions of Cell and SA Aging on the Read-Path Degradation using AppAwareAge

Application-Dependent BTI Impact on SAs and Cells

We investigate the two cases 'SA aging only' and 'combined cell and SA aging' to show the distinct contributions of SA and cell aging to the SD and BS degradation for 3 years of aging at 75°C and nominal supply voltage ($0.9V$).

Fig. 6.2 shows ΔSD (according to equation 4.1) for the case 'SA aging only' (blue bars) and 'combined cell and SA aging' (red bars) corresponding to the applications from Fig. 6.1 for 3 years at 75°C and nominal supply voltage ($0.9V$). The impact of the varying workload resulting from different applications reflects in the SD due to its relatively strong dependency on the the RSP which has been shown in Section 4.3.3. The figure shows that for the case 'SA aging only', ΔSD increases for higher RSP values since the SAs experience more stress. Very unbalanced workloads and the resulting asymmetric V_{th}-shift have a great impact on the degradation, since the symmetry of the design is disturbed. This can be seen very well if we compare applications *Edge* and *Heap*. Application *Edge* shows an RSP that is more than twice as high compared to *Heap*, but ΔSD of

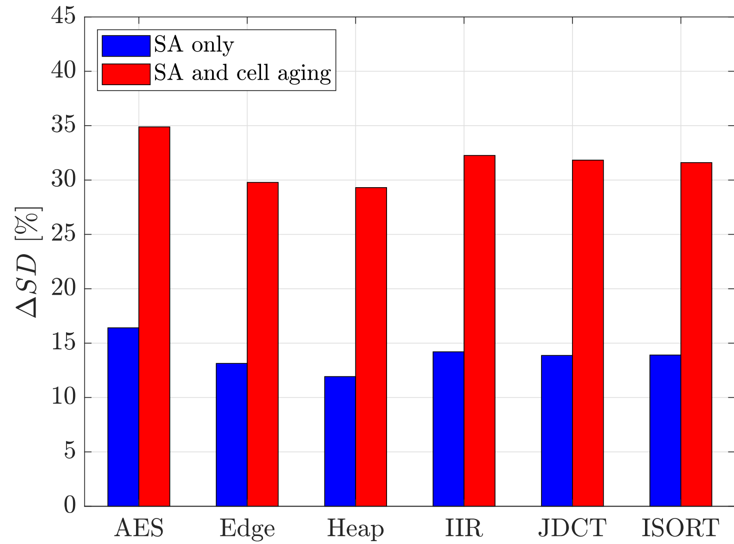

Figure 6.2: Worst-case ΔSD (over all SAs) for various applications - due to SA aging only and combined cell and SA aging

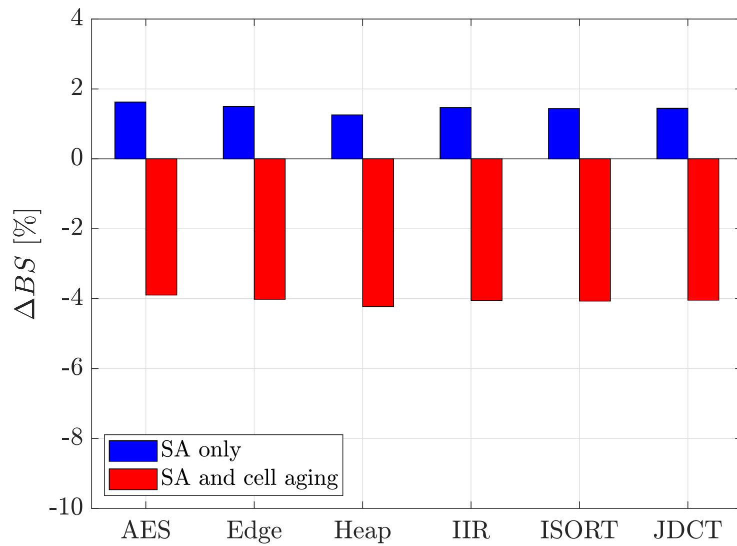

Figure 6.3: Worst-case ΔBS (over all SAs) for various applications - due to SA aging only and combined cell and SA aging

Heap is almost as high as that of *Edge*, because for *Heap* the worst-case SA only reads zeros, causing a very asymmetric degradation. We therefore conclude that workload information is an important factor for precise aging prediction.

For the case 'combined cell and SA aging' it can be observed that the contribution of cell aging to SD results in a much higher ΔSD. This can be explained with the high cell stress compared to the relatively low stress level of the SA. Hence, the effect of cell aging clearly cannot be neglected. For *AES* the wear-out without cell aging is 53.0% lower than for the case where the complete read-path is considered. For *Heap* it is even 59.4% underestimated. Furthermore, the workload dependency and the impact of asymmetric workloads becomes even stronger as can be seen at the example of *Heap* and *Edge* again.

Fig. 6.3 shows ΔBS (according to equation 4.2) for the 'SA aging only' case (blue) and 'combined cell and SA aging' (red). ΔBS generally does not show a large deviation for the different applications since the cell stress is almost equal for all applications and BS is only marginally dependent on the workload of the SA. As already explained in Section 4.3.3, SA aging slightly improves BS due to the later activation of the SA. Hence, SA aging results in a positive ΔBS because the BS after aging is slightly larger than without aging. For the combined aging case, however, it can be seen that this effect is not strong enough to compensate the the contribution of cell aging. Hence, ΔBS is negative (BS is lower after aging than without aging), which translates to an increase in ΔSD since the voltage difference at the input of the SA that needs to be amplified is smaller.

We conclude that the cell aging has a significant impact on the SD degradation and can contribute for more than half of its degradation. This effect results from the fact that the cell stress levels are generally much higher than the SA stress levels while SA seem to be more susceptible and hence lower levels of stress result in a significant degradation. Additionally, for the chosen design the impact of cell aging is very strong and neglecting this effect will cause a considerable underestimation of the wear-out for all considered application workloads. It is worth noting though, that the impact of cell degradation depends on the circuit sizing and strong pull-down transistors in the cell will reduce the impact of cell aging [22]. SA aging slightly improves the BS, however not enough to compensate the strong impact of cell aging on BS.

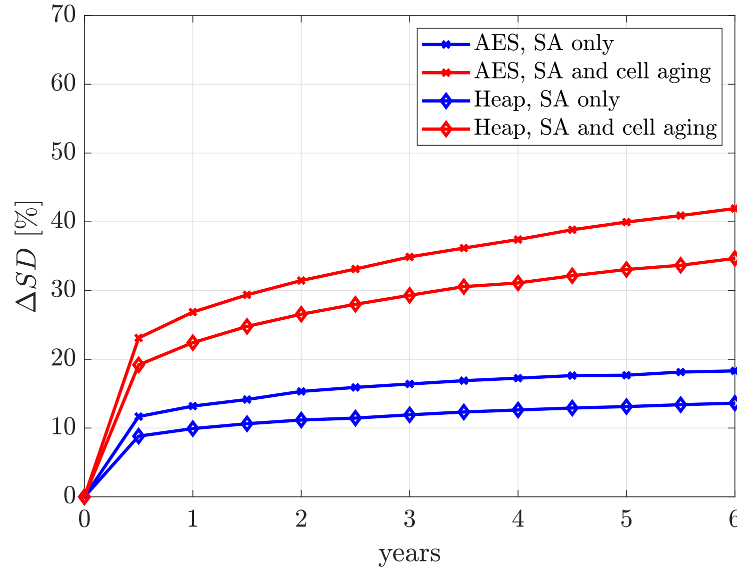

Figure 6.4: ΔSD over Time - due to SA aging only and combined cell and SA aging

BTI Impact on SAs and Cells over Time

ΔSD and ΔBS of the worst-case read-path mentioned before is investigated over time for the best-case and worst-case application workloads for the two cases 'SA aging only' and 'combined cell and SA aging' to illustrate the aging trend over time. We find *Heap* to be the best-case application workload, since it shows the lowest ΔSD with only 29.3% after 3 years aging compared to the nominal time zero SD (see Fig. 6.2). *AES* is found to be the worst-case application workload since it shows the highest ΔSD with 34.9% for the combined case.

Fig. 6.4 shows the ΔSD over time for the worst-case read-path for the application workloads *AES* and *Heap* at 75°C and nominal V_{dd} for the cases 'SA aging only' and 'combined cell and SA aging'. For 'SA aging only', the threshold voltage shift and, hence, ΔSD increases over time. The slope of ΔSD is large in the beginning but flattens over the lifetime of the SRAM, since the threshold voltage shift shows the same behavior.

A similar behavior can be observed for the combined aging case. Compared to the case where only SA aging is considered the maximum degradation for *AES* is underestimated by 56.3% and for *AES* by 60.8% at 6 years of aging. It becomes clear that the impact of cell aging even increases over time.

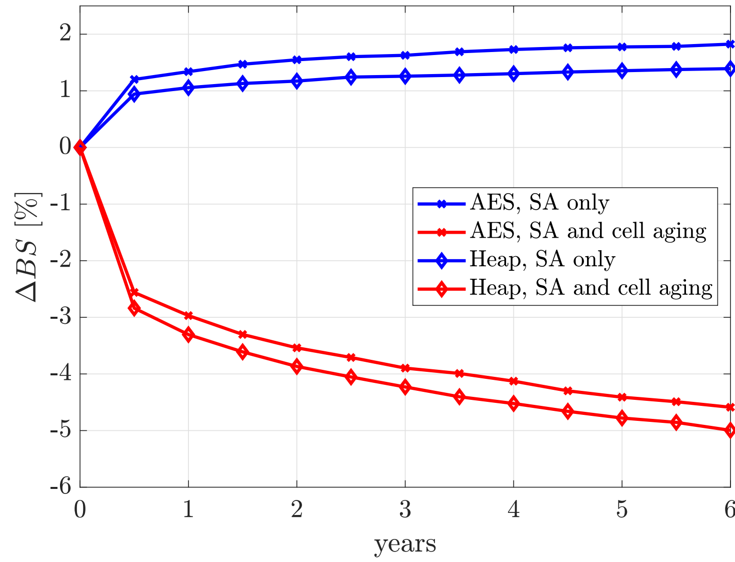

Figure 6.5: ΔBS over Time - due to SA aging only and combined cell and SA aging

ΔBS for 'SA aging only' and 'combined cell and SA aging' is shown in Fig. 6.5. For 'SA aging only', ΔBS is positive and increases over time because of the delayed SA enable. This increase in BS will be overpowered by cell aging in the combined aging case where the resulting ΔBS is negative and decreases over time.

Temperature-Dependent BTI Impact on SAs and Cells

According to the identified worst-case and best-case application workloads, ΔSD and ΔBS of the SRAM array are investigated for three different temperatures for both cases 'SA aging only' and 'combined cell and SA aging' for three years of aging. Figure 6.6 shows ΔSD considering 'SA aging only' and 'combined cell and SA aging' for the application workloads *AES* and *Heap* at $25°C$, $75°C$ and $125°C$ respectively to investigate the impact of temperature on the read-path degradation. For the case 'SA aging only', it is observed that the temperature has a strong impact on the degradation of the SAs. This circumstance is even worse for higher workloads. The degradation is 86.9% higher at $125°C$ than at $25°C$ for *AES*. For the *Heap* application it is 85.3% higher at $125°C$ compared to the degradation at $25°C$.

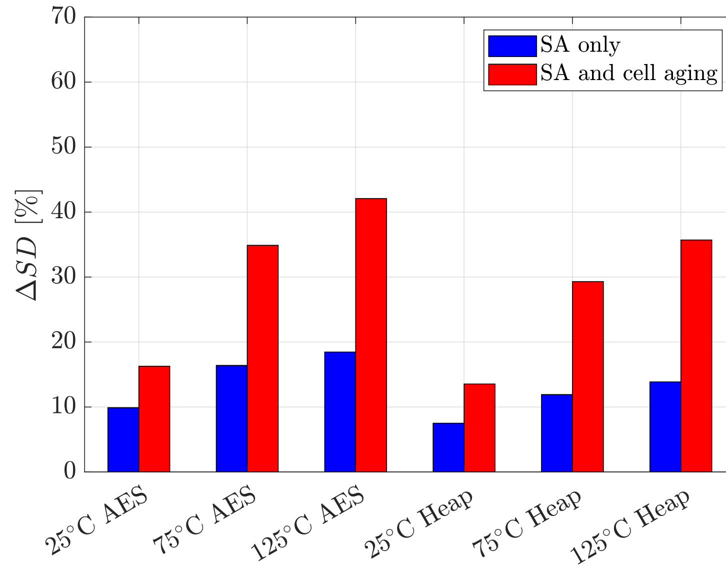

Figure 6.6: ΔSD for different Temperatures - due to SA aging only and combined cell and SA aging

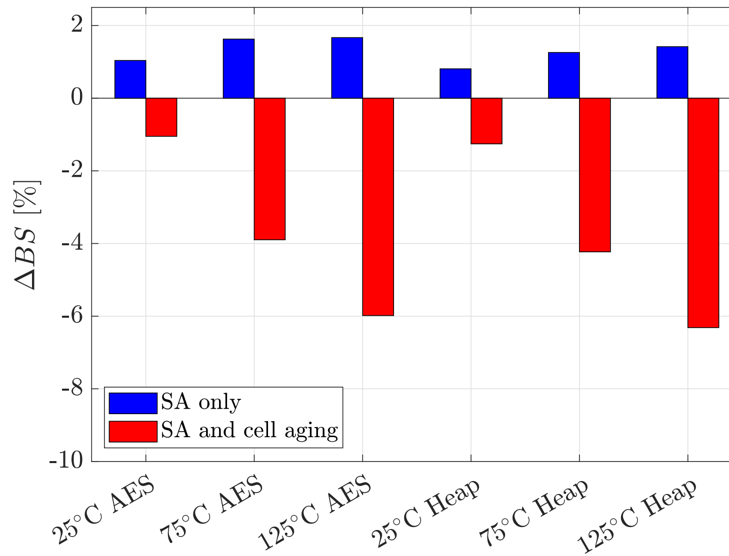

Figure 6.7: ΔBS for different Temperatures - due to SA aging only and combined cell and SA aging

The temperature dependency becomes even stronger for the case 'combined cell and SA aging'. The degradation is up to 158.2% higher at $125°C$ compared to the degradation at $25°C$ for the *AES* application workload. For *Heap* it is 162.5% higher at $125°C$ than at $25°C$.

Temperature also has a strong influence on BS as shown in Figure 6.7. For 'SA aging only', ΔBS is positive and increases at $75°C$ because the SA aging is increased. Interestingly, the ΔBS does not increase much for $125°C$. Since the cell is not affected with BTI in this scenario, BS is only affected by the increase in temperature which slows down the development of BS. Apparently, the temperature effect on BS counteracts the BS improvement caused by larger SA stress at higher temperatures. This is also reflected in the ΔSD which does not degrade by the same amount between $75°C$ and $125°C$ as it does between $25°C$ and $75°C$.

For the combined SA and cell aging the ΔBS improvement is overpowered by the cell aging and ΔBS is negative and steadily degrades. The degradation shows a high dependency on the temperature but only a marginal dependence on the workload.

Supply-Voltage Dependent BTI Impact on SAs and Cells

For the worst-case and best-case application workload, the impact of the supply voltage for three different supply voltages on the ΔSD and ΔBS is analyzed at nominal temperature (25°C) for both cases 'SA aging only' and 'combined cell and SA aging'. Figure 6.8 shows the impact of the supply voltage on the ΔSD at $-10\% V_{dd}$, nominal V_{dd} (25°C) and $+10\% V_{dd}$ respectively. Generally it can be seen that ΔSD is larger for lower supply voltages. This observation is opposing the expectation that a higher V_{dd} is causing a larger degradation but can be explained by the fact that the read-path is much more susceptible at $-10\% V_{dd}$ than at higher supply voltages. Hence adding degradation on top of an already susceptible read-path has a stronger effect than adding degradation on top of the read-path at higher supply voltages, even if the V_{th} shift is larger for a higher V_{dd}. For the case 'SA aging only' it is observed that the supply voltage has a lower impact compared to the temperature. For *AES* the degradation is 26.7% higher at $-10\% V_{dd}$ compared to the degradation at $+10\% V_{dd}$. For *Heap* the degradation is 21.3% higher at $-10\% V_{dd}$ compared to the degradation at $+10\% V_{dd}$. Interestingly, ΔSD does not decrease linearly for higher supply voltages. While the circuit is faster and hence less sensitive at $+10\% V_{dd}$, aging also increases with

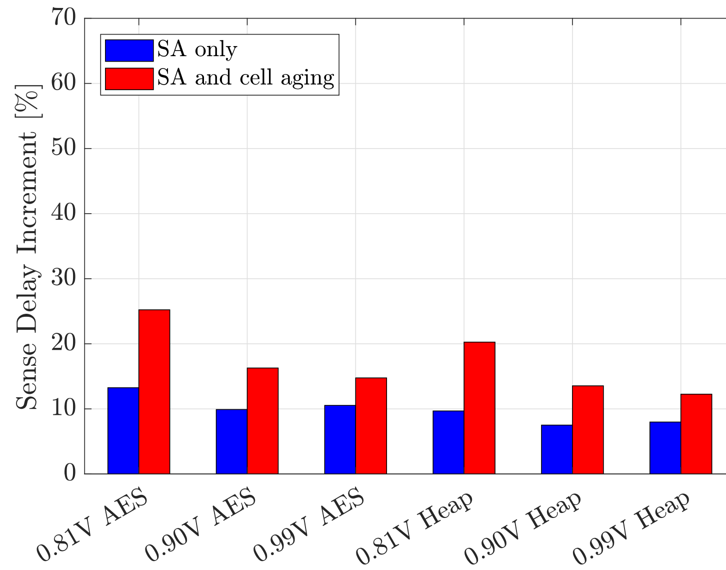

Figure 6.8: ΔSD for different Supply Voltages - due to SA aging only and combined cell and SA aging

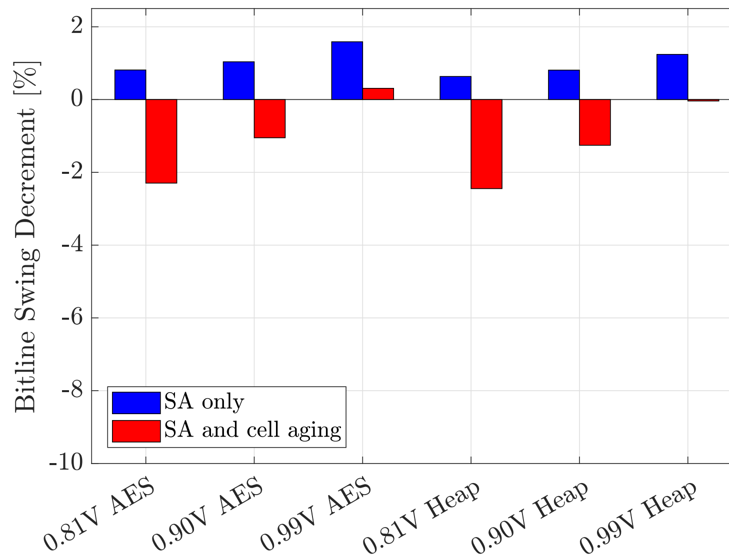

Figure 6.9: ΔBS for different Supply Voltages- due to SA aging only and combined cell and SA aging

higher V_{dd}. These two opposed effects seems to compensate each other to some extent and aging in the SA seems to gradually outpace the effect of a higher supply voltage.

For the combined SA and cell aging V_{dd} has a larger influence. The degradation at $-10\%V_{dd}$ is 70.2% higher for the *AES* application workload compared to the degradation at $+10\%V_{dd}$. For *Heap*, the degradation is 65.0% higher at $-10\%V_{dd}$ compared to the degradation at $+10\%V_{dd}$ and the effect mentioned above cannot be observed anymore.

ΔBS shown in Fig. 6.9 in the case 'SA aging only' is positive and increases because of two reasons: BS improves at higher V_{dd} because the circuit becomes faster and the delayed SA enable signal effectively creates a larger BS. The BS shows a slightly larger increase at $+10\%V_{dd}$ which manifests as a non-linear decrease of ΔSD as explained earlier. Since cell aging is not considered in this case ΔBS is only affected by the supply voltage as SA aging only marginally influences BS.

For the combined SA and cell aging the ΔBS is dominated again by the cell aging and hence negative For the application workload *AES* however, ΔBS remains positive at $+10\%V_{dd}$, because the BS improvement caused by higher supply voltage is so large that even an impaired BS development due to cell aging does not compensate the BS improvement due to SA aging.

6.1.4 Effectiveness of MAGIC

Effectiveness of MAGIC for different Applications

We investigate the effectiveness of MAGIC for all considered applications.

Fig. 6.10 shows the $RSP_{0/1,k,l}$ of the SA of the most stressed read-path (worst-case) of the SRAM memory for different applications with and without the MAGIC aging mitigation. As can be seen, the $RSPs$ of the worst-case SAs are significantly reduced for all application workloads after the application of MAGIC since the read-outs, which were before concentrated in only a few banks, are now distributed to all banks (and hence all SAs) equally. The balancing between $RSP_{0,k,l}$ and $RSP_{1,k,l}$ is as well slightly improved because also data value locations move between banks. Consequently, the worst-case SA is reading the value '1' more often. Hence, some higher portion of $RSP_{1,k,l}$ is visible (light blue) in

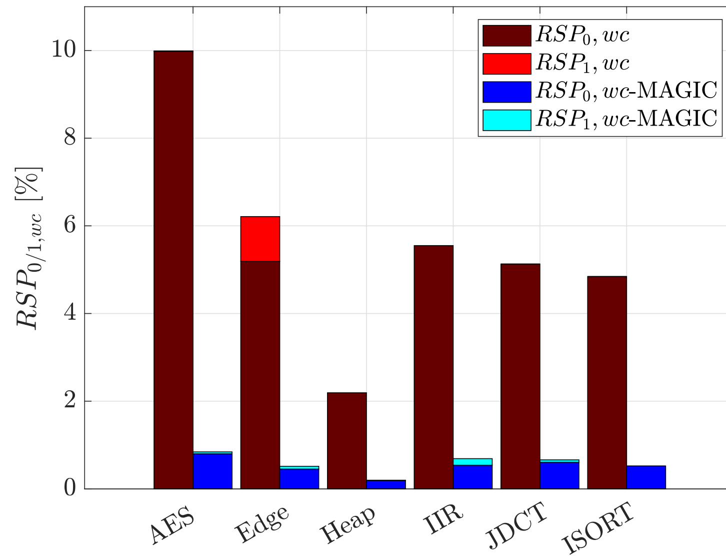

Figure 6.10: Worst-case RSP (over all SAs) for different applications with/without MAGIC aging mitigation

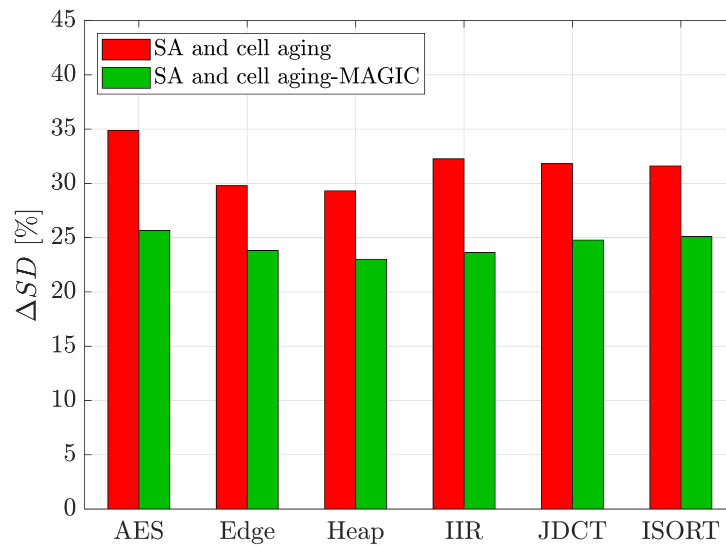

Figure 6.11: Worst-case ΔSD (over all SAs) for various applications - due to combined SA and cell aging with/without MAGIC aging mitigation

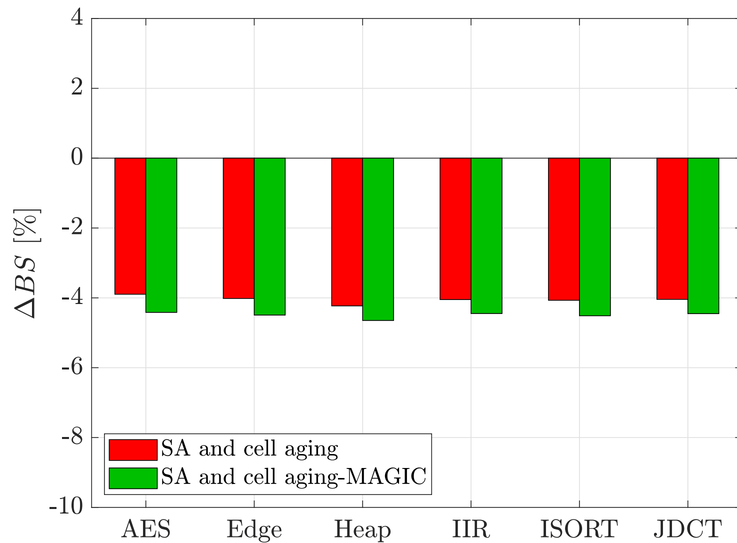

Figure 6.12: Worst-case ΔBS (over all SAs) for various applications - due to combined SA and cell aging with/without MAGIC aging mitigation

Fig. 6.10 for *AES*, *JDCT* and *IIR*. Still, $RSP_{0,k,l}$ remains the dominant factor for all application workloads.

For combined SA and cell aging, the MAGIC mitigation scheme reduces the ΔSD by 26.4% for *AES* and 21.5% for *Heap* as shown in Fig. 6.11 by regularly modifying the bank address.

The cell stress is also slightly mitigated, since the values stored in the cells are moved from one bank to another. However, since a large number of cells are never accessed, they are also not accessed if moved to another bank and, hence, keep their high stress level. Therefore, mitigation of cell aging by MAGIC is not significant for the investigated worst-case read-path and does not reflect in the results. To mitigate cell aging effectively, MAGIC needs to be moved in front of the row or word decoder. This configuration is however not investigated in this work.

Fig. 6.12 shows that ΔBS is more negative, meaning that the voltage difference at the SA inputs is even smaller than without mitigation. This happens since the SA enable signal drivers experience less aging stress and, hence, the SA enable signal is not delayed as much as without the mitigation scheme. Even though the BS reduces (more negative ΔBS), MAGIC obviously still majorly improves the ΔSD as can be seen in Fig. 6.11.

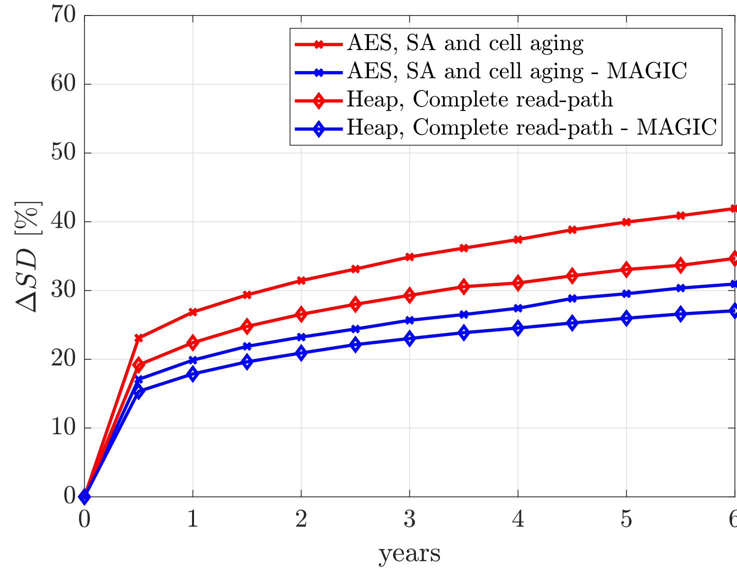

Figure 6.13: ΔSD over Time - due to combined SA and cell aging with/without MAGIC aging mitigation

We conclude that the proposed mitigation scheme can significantly reduce ΔSD up to 26.4% considering SA and cell aging and slightly worsens ΔBS.

Effectiveness of MAGIC over Time

Fig. 6.13 shows ΔSD over time for the worst-case read-path for the workload *AES* and *Heap* at 75°C and nominal V_{dd} for the case 'combined aging' with and without mitigation.

The proposed mitigation scheme significantly reduces ΔSD. MAGIC decreases the degradation about 26.7% for *AES* at 6 years of aging, which significantly improves the lifetime of the memory. For *Heap*, MAGIC improves the degradation by 21.9%. MAGIC is hence very effective for all workloads and gets slightly more effective over time. ΔBS for 'SA and cell aging' without and with MAGIC is shown in Fig. 6.14. As explained before, ΔBS is lower after MAGIC, since the SA aging is reduced.

Effectiveness of MAGIC for different Temperatures

Figure 6.15 shows ΔSD considering 'combined SA and cell aging' for the workloads *AES* and *Heap* at 25°C, 75°C and 125°C respectively to investigate the impact of temperature on the SA degradation.

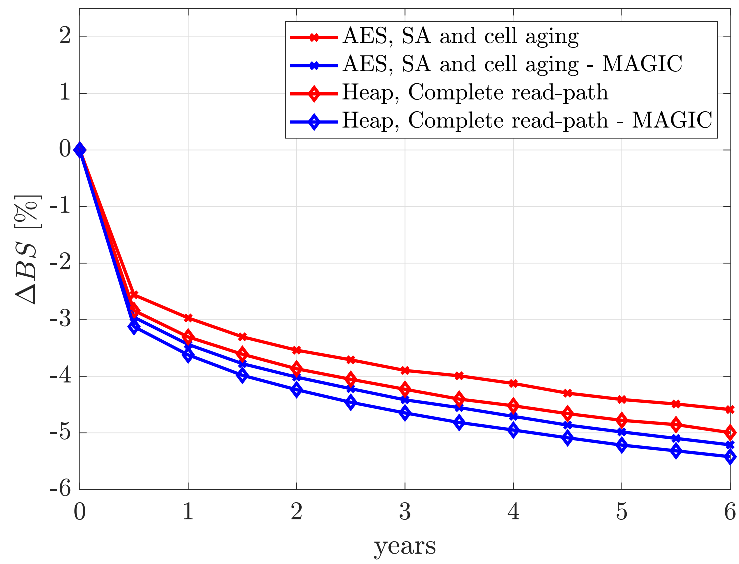

Figure 6.14: ΔBS over Time - due to combined SA and cell aging with/without MAGIC aging mitigation

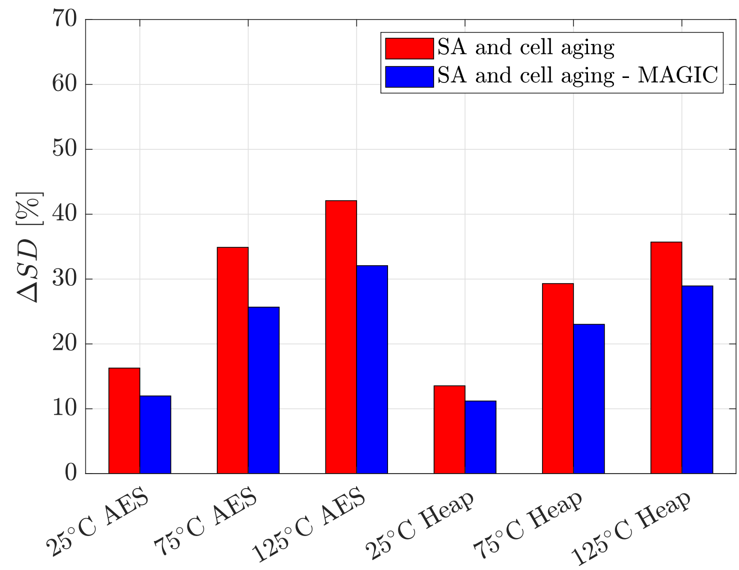

Figure 6.15: ΔSD for different Temperatures - due to combined SA and cell aging with/without MAGIC aging mitigation

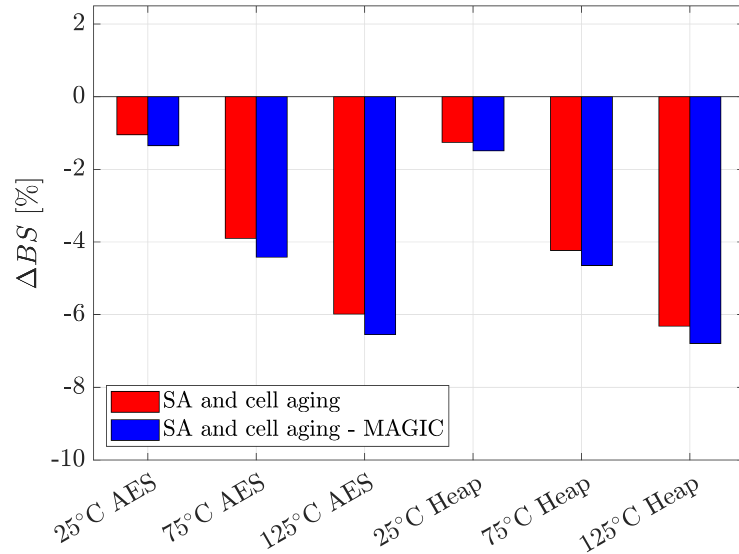

Figure 6.16: ΔBS for different Temperatures - due to combined SA and cell aging with/without MAGIC aging mitigation

The temperature impact on ΔSD after the application of MAGIC is slightly higher than without MAGIC and increases to 167.5% at 125°C compared to the degradation at $25°C$ for *AES* while for *Heap* it slightly decreases to 158.9% at $125°C$ compared to the degradation at $25°C$, still showing a high impact of temperature on the degradation. ΔBS is shown in Fig. 6.16 for the combined case. ΔBS steadily becomes more negative for higher temperatures, since aging in the cells gets worse over temperature. Again, MAGIC decreases the bit line swing marginally since aging in the SA enable signal drivers decreases.

Effectiveness of MAGIC for different Supply Voltages

Figure 6.17 shows the impact of the supply voltage on ΔSD at $-10\%V_{dd}$, nominal V_{dd} and $+10\%V_{dd}$ respectively at nominal temperature (25°C).

The supply voltage dependence of ΔSD reduces for *AES* if the mitigation scheme is applied since the SA workload is significantly reduced. For *AES* the supply voltage dependency reduces to 60.6% at $-10\%V_{dd}$ while for *Heap* it slightly increases to 66.8% at $-10\%V_{dd}$ compared to $+10\%V_{dd}$. ΔBS steadily decreases, hence becoming less negative for higher supply voltage since the circuit becomes faster as shown in Fig. 6.18. ΔBS degrades more after the application of MAGIC

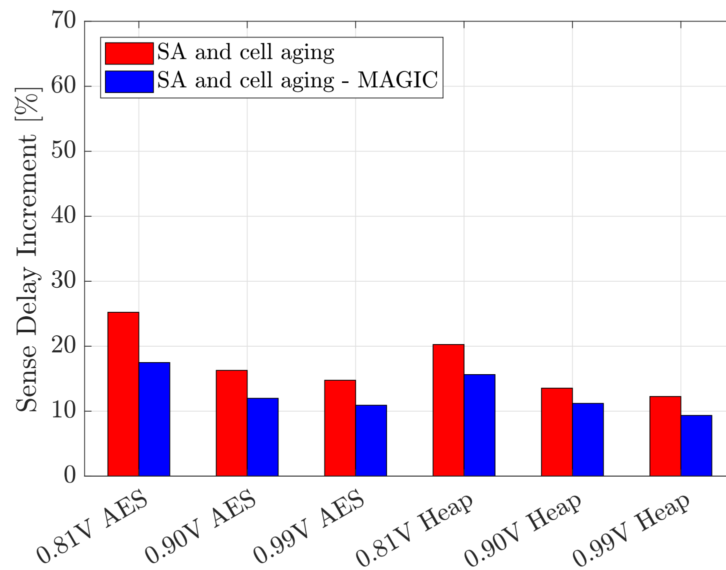

Figure 6.17: ΔSD for different Supply Voltages - due to combined SA and cell aging with/without MAGIC aging mitigation

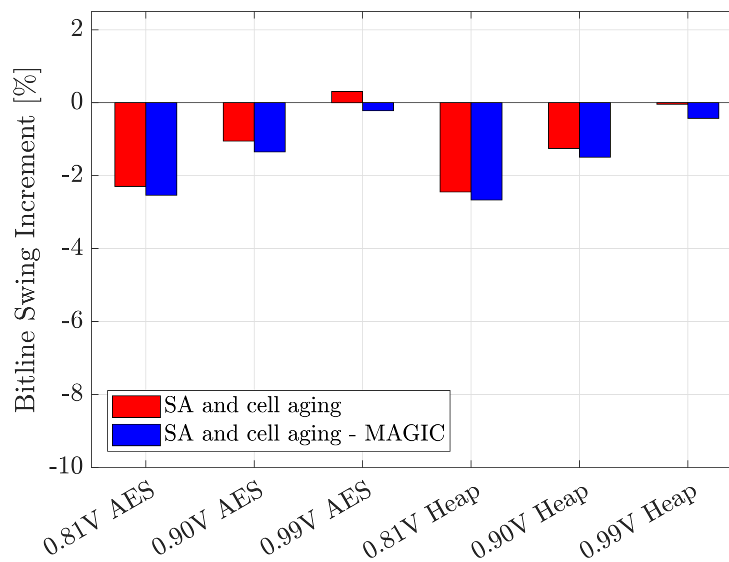

Figure 6.18: ΔBS for different Supply Voltages- due to combined SA and cell aging with/without MAGIC aging mitigation

for both application workloads. This also leads to the effect that ΔBS now stays negative even for the case *AES* at +10%V_{dd} since the SA enable drivers age less.

Effectiveness of MAGIC for Workloads from Multiple Application Tasks

The applications that are investigated in this work ran individually on the MCU. If the MCU is running a set of application tasks, each task should be profiled with the binary compiled with all tasks and, possibly, the runtime system because the memory layout, e.g., the data section and stack accesses, may be different compared to running the task using an individual binary. Different task schedules can be taken into account by constructing a combined memory trace from the single task traces by making sure the trace reflects the tasks' periodicity. Additionally, time stamps can be adapted to reflect inactive times of the MCU, e.g., for application with sleep or power down phases, which would enable the investigation of aging recovery strategies.

In this work, only six individual application tasks were investigated. All applications showed a similar behavior: addresses which correspond to often used data inside the data section or the stack of the executed program are accessed more frequently. In particular, the worst-case stress is mostly located in the stack section of the program. Memory addresses between stack and static data section are often used less frequently or not at all since many programs do not make full use of the available memory. Stack and static data section are often located near the beginning and at the end of the address range. This is of course only valid under the assumption that the system runs without OS or MMU.

If tasks share the same stack section, the stress of all tasks is combined there and the MAGIC mitigation scheme to distribute the stress works as proposed. If the runtime environment such as the Real-time Operating System (RTOS) moves or assigns different stack frames to tasks, then the effect of additional stress in the stack section does not hold. In the worst case, the layout of the stack frames could counteract the mitigation when stress is only moved between stack frames. Yet, as a major advantage of the proposed method, this would be identified through the analysis with the AppAwareAge tool and a different assignment of stack frames can be used in the RTOS to allow effective aging mitigation. Hence, in order to accurately estimate the lifetime of a specific memory we want to stress that for different workload conditions the analysis proposed in our work needs to be repeated with a new memory trace (e.g. for running different/several applications) which is easily possible with the proposed framework.

6.1.5 Lifetime Prediction without and with MAGIC

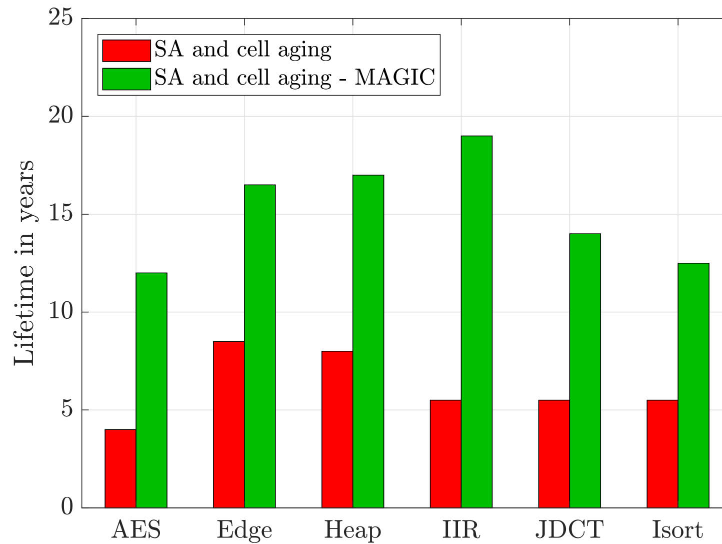

Figure 6.19: Lifetime of SRAM for various workloads

After defining the end-of-life for the memory using equation 4.9, the expected lifetime of the SRAM is predicted for 'combined cell and SA aging' with and without mitigation for all application workloads and a harsh use case of 90°C and −5% V_{dd} for the combined aging case. We predict the lifetime of the SRAM array using equation (4.9) with f_{safety} = 1.3 to add timing margin to the measured SDs in order to account for additional process variations which are not incorporated in the use case.

As shown in Fig. 6.19 for the application workload *AES*, the memory exceeds the given margin and hence reached its end-of-life already after 4 years without the mitigation scheme, since the given use-case is very harsh. After the application of the proposed mitigation scheme we observe a memory lifetime of 12 years. Accordingly, the memory has been protected from exaggerated aging stress with MAGIC and can achieve a 3x longer life even for harsh operating conditions. *Heap* reached its end-of-life after only 8 years without the mitigation scheme. Here, MAGIC can improve the lifetime of the memory to 17 years, which corresponds to an improvement of 2.1x.

Unfortunately, this work neglects process variations due to simulation time restrictions as simulating a whole memory array with Monte Carlo is costly. However, we recognize that variations and mismatch play a significant role in SRAMs

due to the fact that they are designed for highest integration densities and continue to lead the migration to new technology nodes. To account for process variations we introduce f_{safety} to add timing margin to the measured sensing delays since the assumption of the end-of-life of a circuit at -10% at 125°C V_{dd} while not taking into account variations seems to be an unrealistic condition. Hence, we assumed 20% additional margin for process variation and mismatch. Adding another 10% seemed reasonable for the investigated design since for the given operating frequency there is still timing margin available before the read-outs in the memory actually begin to show errors. It should be clearly state though, that this f_{safety} is only a tuning parameter used for post-processing of the simulation results and can easily be adjusted without repeating any simulation. It can and should be adjusted individually to a value appropriate to the investigated design and operating conditions, especially since the available margin clearly depends on the design and the operating frequency.

6.1.6 Area Overhead

Since the memory configuration in our example only has 16 banks, the bank address only has 4 bits. To be able to calculate the area of the MAGIC block we assume a volatile counter that has its own supply voltage which stays high, if the system is powered up/down. A non-volatile counter can be applied just as well, with the benefit, that it holds the value without any additional power supply. An example would be an MCU with a FLASH memory that can be modified by the chip.

Without loss of generality, we assume a 4-bit synchronous counter composed of D flip-flops to calculate the area overhead. MAGIC only has to be inserted once before the decoder stage, hence the additional area is restricted to:

Table 6.1: Area overhead of MAGIC

	Transistors per Component	Total transistors
XOR	6	24
D flip-flop	16	64
Total	-	88

6.2 SRAM Design Exploration (SDE)

6.2.1 Experimental Setup

In this section we present our experimental results obtained from the aging-aware SDE for a 64kByte On-Chip memory of an OpenRisc 100 processor (OR1k) [98] in 32nm technology [96]. We ran a representative set of applications on the processor to obtain average *RSP*s for each SA. As our use-case we chose workloads from 15 applications including sorting algorithms (ISORT, HEAP), image processing and compression (EDGE, JDCT), encryption algorithms (AES), digital filter algorithms (IIR, FIR) and several arithmetic computations. The applications are run consecutively and the results represent the average wear-out caused by the 15 applications running on the processor for an extrapolated lifetime of 3 years of aging at 75°C.

6.2.2 Stress Profiles of all possible Memory Configurations

The respective memory accesses of this use-case can be seen in Figs. 6.20 and 6.21, which show the number of reads and writes for each address (in decimal), respectively. Here, the observation that most read and write traffic is happening upon heap and data section near the beginning and at the end of the address range is confirmed.

For the aging-aware SDE we chose the memory granularity configurations as shown in Table 6.2. The table contains the resulting number of banks for a given combination of rows and columns for a memory size of 64kByte. Hence, e.g. 64x64x4 indicates the memory configuration containing 64 banks, 64 rows and 4 words.

Table 6.2: Memory Configurations

		Number of words			
		4	8	16	32
Number of rows	64	64	32	16	8
	128	32	16	8	4
	256	16	8	4	2
	512	8	4	2	-

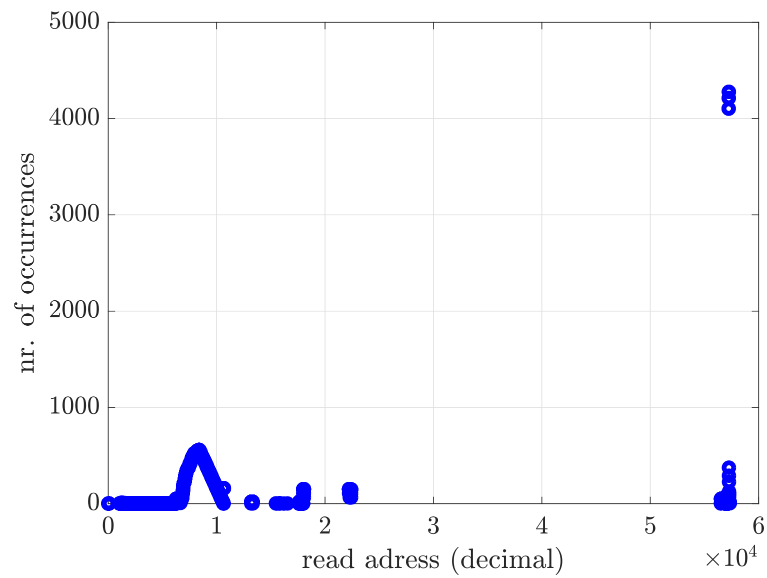

Figure 6.20: Read Addresses of representative Set of Applications

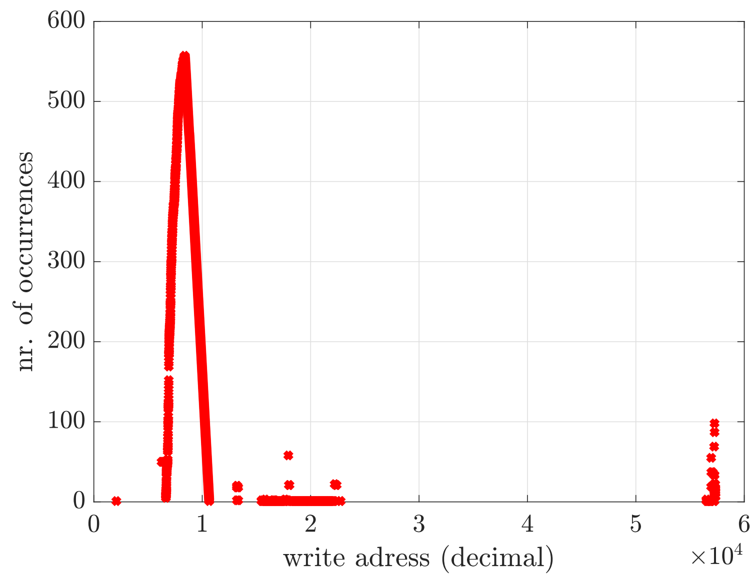

Figure 6.21: Write Addresses of representative Set of Applications

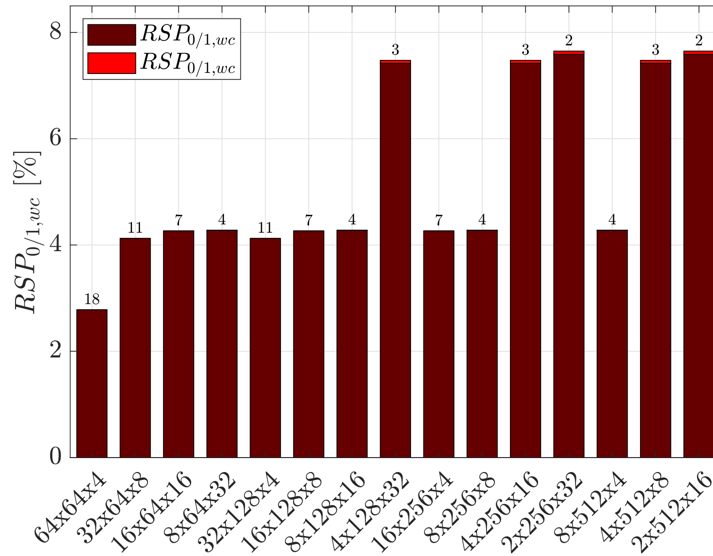

Figure 6.22: Worst-case RSP (over all SAs) over all Granularity Configurations

Figure 6.22 reveals the resulting $RSP_{0/1,wc}$ of the SA of the most stressed read-path (worst-case) of the SRAM memory for the different memory granularity configurations as specified in Table 6.2. The column of this read path would potentially be the first one to fail completely. Each bar represents one of the configurations and shows both $RSP_{0,wc}$ and $RSP_{1,wc}$ as stacked bars. Furthermore, the figure shows the number of banks that the applications access for each granularity above each bar. As expected, the $RSP_{0/1,wc}$ is decreasing for a higher number of banks because the workload is spread across more banks. This can be seen particularly well if we compare the RSPs of the configurations 64x64x4 and 2x512x8. Figures 6.23 and 6.24 exemplarily show $RSP_{0,wc}$ and $RSP_{1,wc}$ of the complete SRAM array for a granularity of 64x64x4 (64 banks, 64 rows and 4 words). The workload of the frequently accessed addresses for stack and static data section is distributed to 18 out of 64 banks. Figures 6.25 and 6.26 shows $RSP_{0,wc}$ and $RSP_{1,wc}$ respectively for a granularity of 2x512x8 (2 banks, 512 rows and 8 words). The frequently accessed addresses decode to 2 out of 2 banks, which illustrates the effective reduction of RSP for arrays with a higher number of banks.

Configurations with the same number of banks show the same amount of degradation. This makes sense, because the number of bank bits to select the corresponding banks is the same and hence the decoding does not change. For the same number of banks the individual banks must contain the same number of

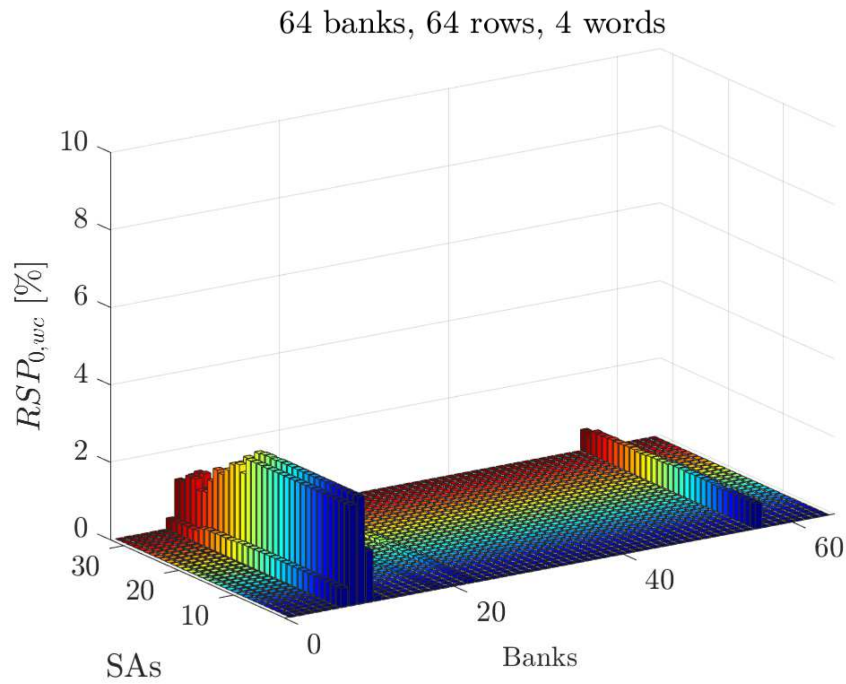

Figure 6.23: RSP_0, 64 banks, 64 rows and 4 words

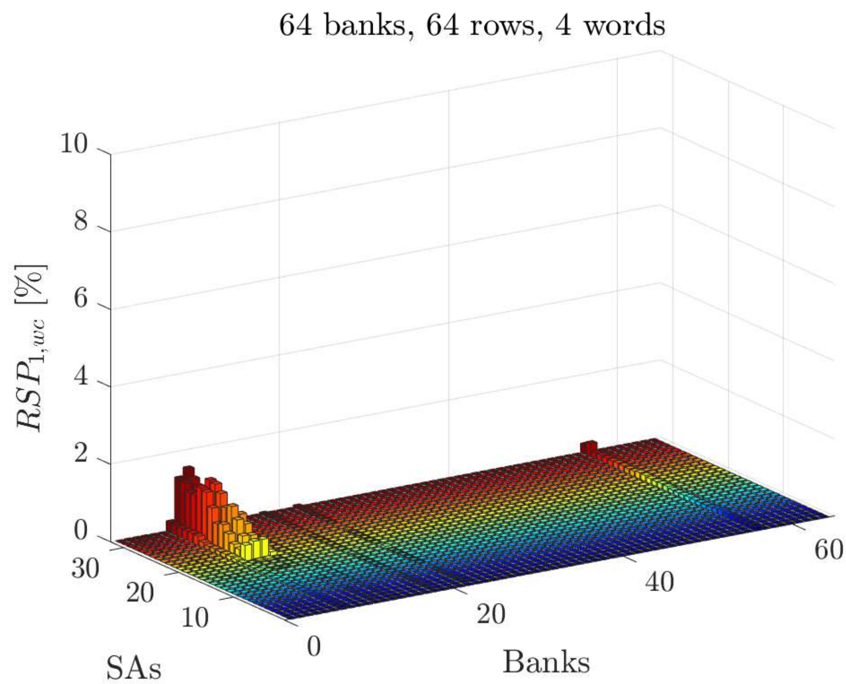

Figure 6.24: RSP_1, 64 banks, 64 rows and 4 words

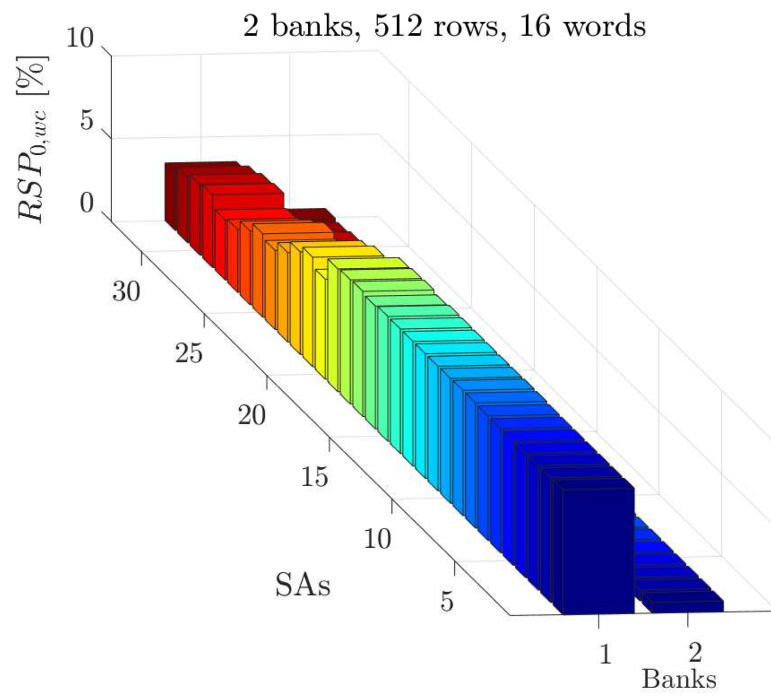

Figure 6.25: RSP_0, 2 banks, 512 rows and 16 words

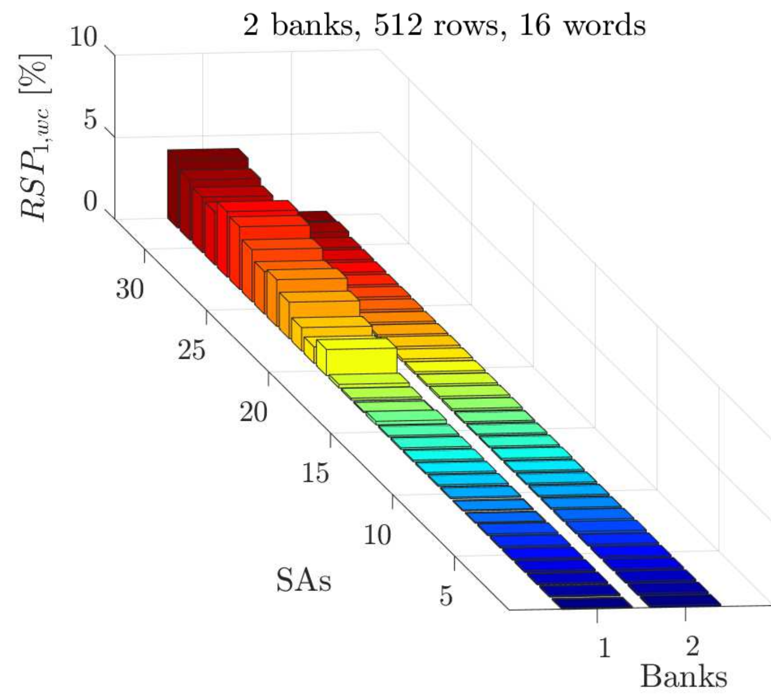

Figure 6.26: RSP_1, 2 banks, 512 rows and 16 words

words, no matter what configuration regarding the number of rows or columns is chosen. All words in a bank use the same set of SAs. Furthermore, it can be observed that all configurations with 32, 16 and 8 banks show a quite similar $RSP_{0/1,wc}$ since the number of banks the workload decodes to is not changing significantly (4 to 11 banks are accessed) and hence the stress that the worst-case read-path experiences does not change notably. Only for the configurations with 64 banks a significant reduction in the $RSP_{0/1,wc}$ can be observed (access to 18 banks). The $RSP_{0/1,wc}$ increases between the configurations with 4 and 2 banks although the accessed number of banks only decreases by one if the configuration is changed from 4 to 2 banks. This can be explained by the fact that that these configurations contain significantly more words per bank (either per row or because the number of rows significantly increased). Since each word is using the same sub-set of SAs, the worst-case read-path in this bank consequently experiences much more aging stress which hence results in a higher $RSP_{0/1,wc}$. The numbers for the $RSP_{0/1,wc}$ shown in this section are generally lower compared to the $RSP_{0/1,wc}$ from 6.1 since we are investigating the average worst-case $RSP_{0/1,wc}$ over all applications compared to the overall runtime of all 15 applications. For the chosen set of applications again the value '0' is read significantly more often in the worst-case read-path here than '1' as already explained in Section 6.1.2, which clearly results in asymmetric aging of this SA.

6.2.3 Read-Path Degradation of all possible Memory Configurations

To predict ΔSD of the chosen granularity configurations we applied a use-case of 3 years of aging at $75°C$ to calculate the threshold voltage shift due to BTI aging. The simulations to measure the SD were conducted reading out the value '0' at the top cell in the array, since this represents the worst-case read-out condition, as the read-path is already experiencing a high stress from reading the value '0'. This represents the worst-case read-out condition since the top cell drives the parasitics of the whole bit line. Figure 6.27 shows ΔSD for the considered configurations from Table 6.2 considering aging of the complete read-path. ΔSD increases for higher RSP values since the SAs experience more stress. Workloads which are a more balanced in terms of reading not only the value '0' result in a less asymmetric V_{th}-shift and greatly improve the degradation, since the symmetry of the design is disturbed less. This can be seen very well for configuration 4x128x32 which has an RSP that is almost twice as high compared to other con-

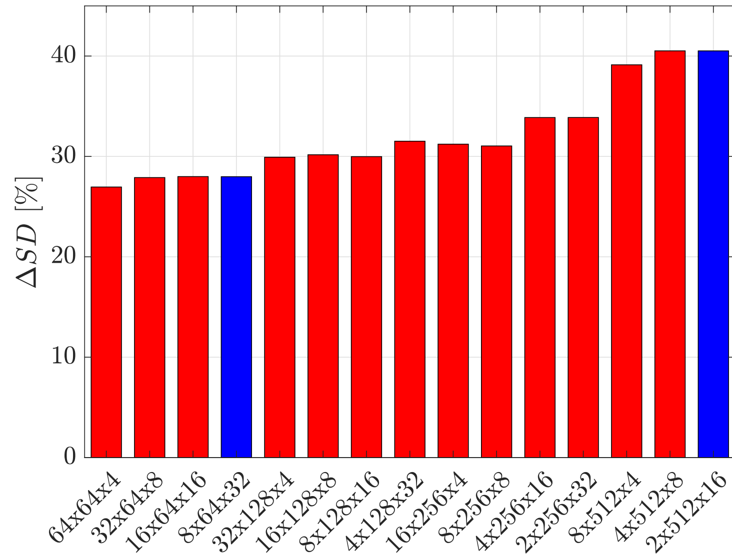

Figure 6.27: Worst-case ΔSD over all Granularity Configurations

figuration like e.g. 8x128x16 but does only result in a moderately higher ΔSD, since the shown worst-case SA is also reading the value '1' during its lifetime. This is also true for all other configurations where ΔSD does not increase with higher $RSP_{0/1,wc}$ however the $RSP_{1,wc}$ for many configurations is too small to be noticable in the bar plot. It is observed, that more rows lead to a higher degradation. This makes sense, because the longer the bit line, the more capacitance is attached to it. Notably, this effect even has a stronger impact on ΔSD than the number of banks for the assumed workload.

Adjusting the memory granularity to contain more banks spreads the appearing stress over additional banks and reduces the stress probabilities of all banks. Hence, increasing the number of banks can prevent specific SAs from failing completely, thereby increasing the lifetime of the memory. Additionally, choosing a configuration with a lower number of rows can improve the sensitivity and hence degradation of the memory. Since aging is dependent on the workload and hence the application, an individual study is required for a different set of workloads. From Fig. 6.27 we chose 8x64x32 as the best-case granularity because the maximum degradation is only slightly higher compared to the configurations with more banks (16x64x16, 32x64x8 and 64x64x4) while invoking less area penalty. We compare it to 2x512x8 because it shows the worst-case degradation (blue bars). While 2x512x8 already reaches a degradation of 40.5%, 8x64x32 shows a maximum degradation of only 28.0%. Hence, the proposed

SDE framework can mitigate aging in the SAs up to 31.6% merely by choosing the appropriate array granularity.

6.2.4 Area Overhead

For the chosen representative set of applications this means that choosing a granularity of 8x64x32 effectively mitigates aging and improves the lifetime of the memory array significantly. The area penalty however is an additional 6xL SAs with L being the word size compared to the worst-case granularityof 2x512x8. For $L = 32$ this would mean an area penalty of 192 SAs which for the designed used in this work corresponds to 3264 transistors.

6.3 Runtime Power, Temperature and Aging Monitoring on FPGA Prototypes (eTAPMon)

6.3.1 Experimental Setup

In this section we present our experimental results obtained from the eTAPMon running an exemplary operating scenario. A 2x2 multi-core design is assumed for the monitored prototype. The FPGA prototype runs at 100MHz and delivers data characterized from a target ASIC LEON3 processor running at 1200MHz at nominal process corner. The monitoring period is set to 1ms on the FPGA which corresponds to approximately 83μs on the target ASIC. The presented data for power, temperature and aging was obtained directly from the FPGA platform via the user serial port and an additional debugging UART USB bridge. The selected operating scenario sequentially distributes tasks first to core 1, then to core 2, 3 and 4.

6.3.2 Emulation Results from the Implementation of eTAPMon on an FPGA Board

Fig. 6.28 shows the power monitor outputs of the four blocks IU, RF, D Cache and I Cache for all four cores for the runtime of 42s. Core 1 is the first core

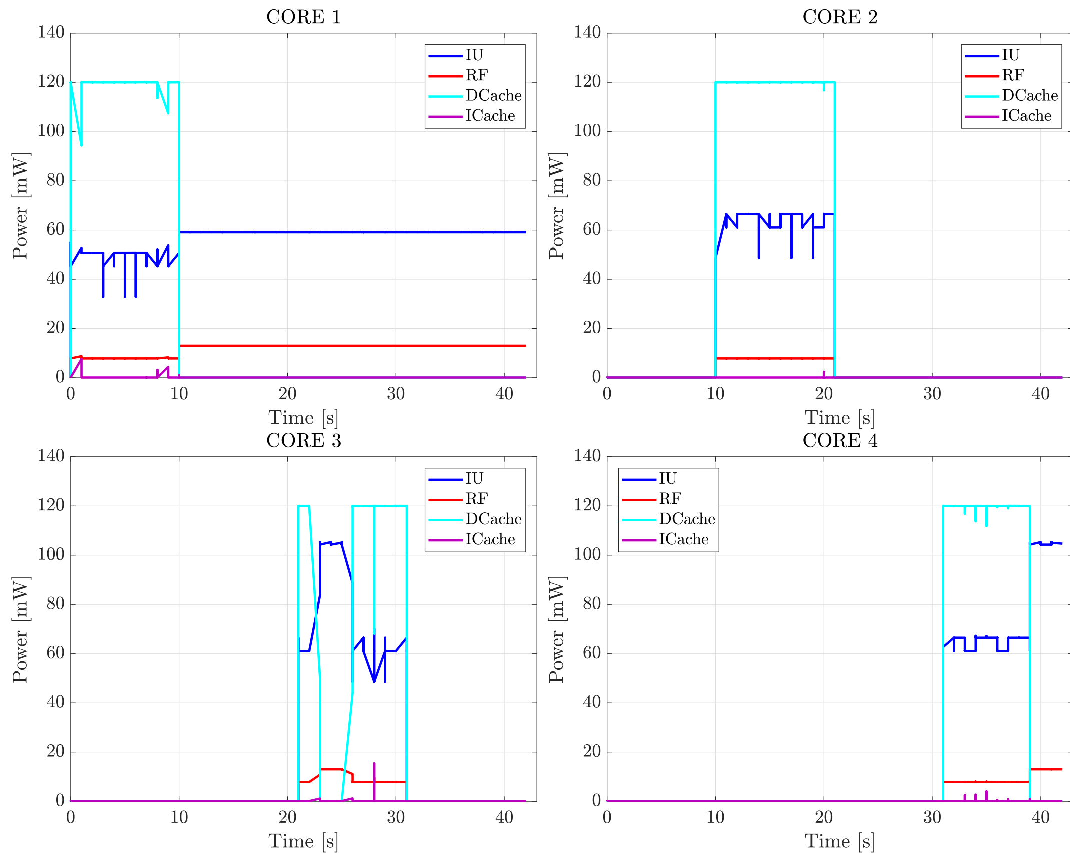

Figure 6.28: Power Monitoring of the Four cores.

that is activated. This core has a special role in the LEON system. It has to be always on and is therefore trapped in a while(1) loop, which means that it will always consume a certain amount of power even after finishing its task. The remaining cores 2, 3 and 4 are set to inactive, when they run no task and hence do not consume any power when de-activated. Since the task is run sequentially on each core (starting from core 1, ending at core 4) the scheduler also only activates the cores sequentially as can be seen in the power diagrams of the four cores. Furthermore, it can be observed that for the chosen operating scenario the largest power contributors are the D Cache and the IU.

Figure 6.29 shows the output of the temperature monitor of the system over the runtime. In the beginning, when only core 1 is running a relatively low temperature can be observed. As soon as core 2 starts up, the temperature rises since core 1 is not going to standby like the other cores. Hence, it still consumes power as can be seen in Fig. 6.28 thereby increasing the overall power consumption.

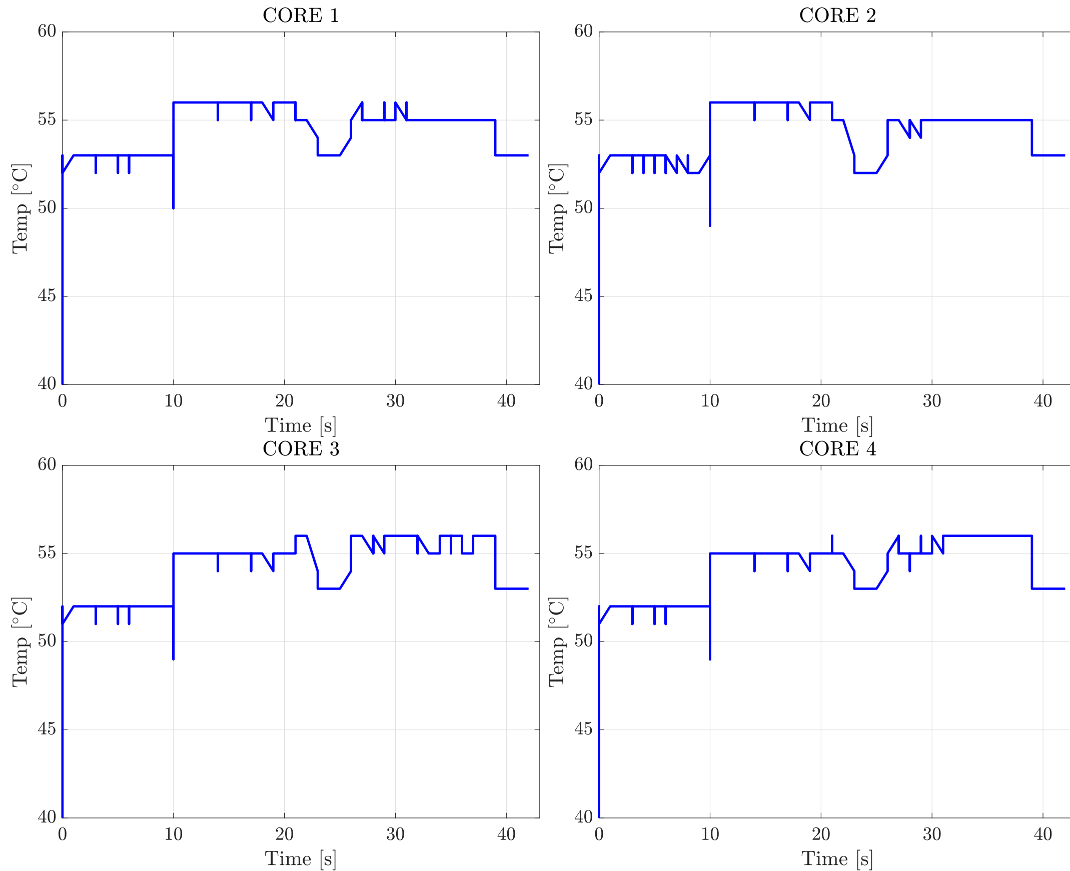

Figure 6.29: Temperature Monitoring of the Four cores.

As soon as core 3 is activated, core 2 is de-activated and hence the temperature behavior is resembling the scenario for core 2. The same can be observed for the active time of core 4. The temperature change is relatively moderate. During operation the temperature varies between 52 and 56 degrees, which is due to the fact, that only one core at a time is activated in our scenario. All four cores show a very similar temperature behavior meaning that the neighbor effect is very strong between the cores for the given floorplan.

Figure 6.30 shows the results for the slack from the aging monitor in the IU. Since the system is only running for a few seconds, no aging is occurring. Hence, the slack only varies slightly with temperature as shown in Fig. 6.29. To predict the degrading frequency margin of the IU for the given workload, Fig. 6.31 shows the same operating scenario while the aging acceleration described in 5.3.3 is turned on. The runtime of the operating scenario is extended to 482 days as observed by the aging monitor. Hence, V_{DD} and T are not changing in real-

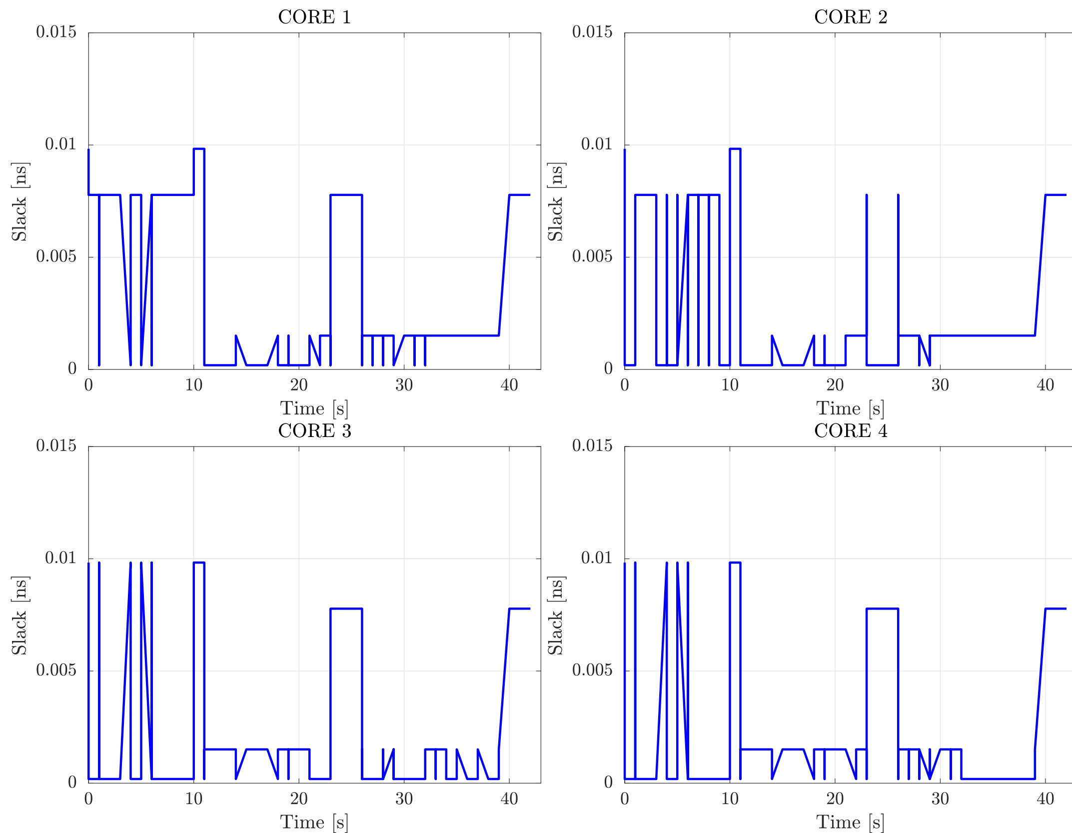

Figure 6.30: Slack Monitoring of the Four cores for Nominal Corner.

time anymore but within days. This acceleration adds inaccuracy in terms of the aging prediction but comes close to a scenario for which the operating scenario would be repeatedly executed for this extended time. Aging accumulates much faster, hence reducing the slack until it becomes negative, indicating that the timing might get violated. The fast changing fluctuations of the monitored slack are primarily due to the change of the chip temperature, the overall trend of the curve relates to the degradation due to aging. Utilizing this monitoring data, the runtime power manager could prevent possible failures, e.g. by switching to a lower frequency before the monitored slack becomes negative.

6.3.3 Hardware Overhead

Table 6.3 summarizes the hardware overhead invoked by the implementation of eTAPMon to monitor a 2x2 Leon3 multi-core system. It shows the number of LUTs, FF (Flip-Flops) and DSPs (Digital Signal Processing Units) that each of the

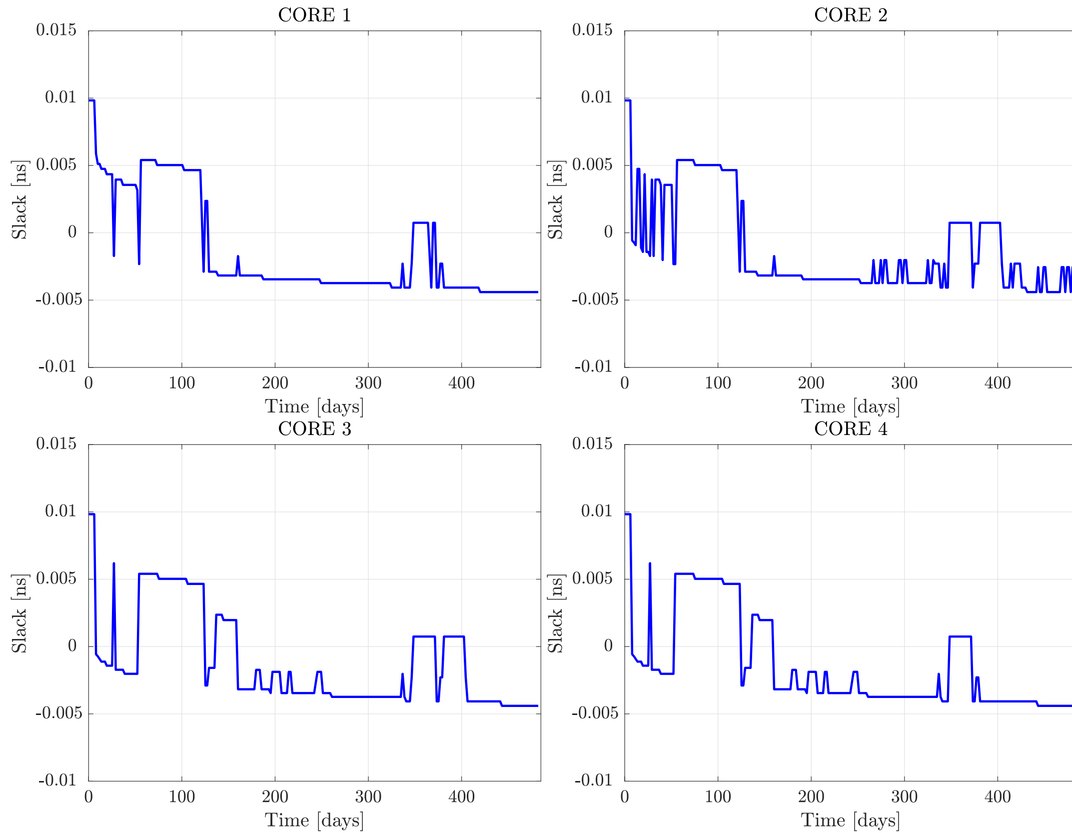

Figure 6.31: Slack Monitoring of the Four cores with accelerated Aging.

monitors requires for the implementation on FPGA. Furthermore, the hardware overhead for the APB communication interface and the overall hardware utilization of the Leon3 Quad Core including a floating-point unit is summarized in the table. For the considered 2x2 multi-core design, the power monitor displays a hardware overhead of 1.02 % LUTs, 3.08 % FFs and requires additional 7 DSPs. The temperature monitor utilizes 0.99 % more LUTs and 6.63 % more FFs and one additional DSP. The aging monitor shows the highest hardware utilization despite the simplified BTI Model and hence shows a hardware overhead of 13.61 % LUTs and 6.61 % FFs. The APB Interface adds additional 0.23 % LUTs and 0.25 % FFs and therefore adds an almost neglectable overhead. The total hardware overhead for the eTAPMon system hence results in 15.85 % more LUTs, 16.58 % more FFs and 8 additional DSPs.

Table 6.3: eTAPMon Hardware Overhead

	LUT	FF	DSP
Power Monitor (per core)	268	400	7
Temperature Monitor	1046	3442	1
Aging Monitor (per core)	3581	858	-
APB Interface	237	131	-
Leon3 Quad Core mit FPU	105220	51913	-

6.4 Summary

This Chapter introduces the experimental results obtained from the novel reliability tool for SRAM Design-for-Reliability where we analyze the contributions of SA and SRAM cell aging for various application workloads, temperatures and supply voltages. Our results show, that realistic workloads are a vital factor for an accurate aging prediction. Furthermore it is shown that cell aging has a large impact on the SRAM degradation and cannot be neglected, while SA aging slightly improves the *BS* however not enough to compensate the strong impact of cell aging.

Furthermore, the results for the first SRAM aging mitigation technique MAGIC are presented to demonstrate its effectiveness. The proposed mitigation scheme can significantly improve the degradation of the SRAM, which is extremely important for safety-critical systems to guarantee full functionality till the end of life of the device. Moreover, this Chapter summarizes the results obtained for the second SRAM aging mitigation technique SDE. The presented results show that the array granularity has a significant impact on the aging behavior of the memory. Hence, SA aging can be efficiently mitigated by choosing the most favorable array granularity in terms of aging. Therefore, the proposed design exploration can be a helpful means during the design phase of safety critical systems to predict the memory lifetime and mitigate aging by selecting the most reliable design for the intended set of applications.

Finally, the results of the real-time eTAPMon monitoring system are discussed. The monitor system was implemented on an FPGA board and evaluated for a selected operating scenario for nominal process corners, where it provides useful insights on the power, temperature and aging behavior of the system.

7 Conclusion and Future Work

Continuous technology scaling has provided a steady increase in the processing power of Integrated Circuits (ICs) over the past decades. On the downside, higher power dissipation, rising on-chip temperatures and an increasing impact of time-zero and time-dependent transistor wear-out mechanisms put product reliability at high risk. Hence, optimizing on-chip systems for reliability is nowadays an equally important design goal as optimizing them for performance and yield. Especially in safety-critical systems, high reliability requirements dramatically increase the costs to ensure an operation within an acceptable error rate and prohibit a loss of operation before the product reaches its intended lifetime. With higher system complexities and circuit sensitivities, reliability assessment during the design phase of the product is more important than ever to reduce costs and guarantee that reliability specifications are met. The scope of this thesis is therefore to predict, mitigate and emulate arising reliability threats caused by power consumption, temperature and aging in two major building blocks of modern SoCs: the on-chip SRAM and the Central Processing Unit (CPU).

One important contribution of this work is a novel reliability tool for SRAM Design-for-Reliability incorporating a new workload-aware aging analysis for On-Chip SRAMs. The tool AppAwareAge predicts the aging-induced degradation of on-chip SRAMs based on the workload of embedded applications executed on an automotive and industrial MCU while considering aging in the complete read-path and its control signals. It therefore accurately captures BTI stress and recovery phases as they appear in realistic workloads and can hence provide a useful means for reliability assessment early in the design phase. An extensive study for an industrially used SRAM design analyzes the proposed workload-aware aging analysis for various workloads, temperatures and supply voltages and reveals the significant contribution of SRAM cell and SA aging to the degradation of the read-path. Another interesting outcome is that aging effects inside interacting sub-components of the read-path are not necessarily magnifying each other. Aging in the SA's control signals counter-intuitively leads to minor performance improvements. This underlines the importance to

analyze the aging behavior of the complete memory block to avoid expensive safety margins without sacrificing reliability. Furthermore, an application-aware end-of-life analysis of the SRAM can predict the expected lifetime for the given workload and operating conditions. Since the tool builds a bridge between high-level and low-level simulation methods it is able to accurately characterize aging in on-chip SRAMs with low computational effort. The physical BTI model is separated from the circuit analysis to enable an efficient integration of updated models for new technologies. Chapter 4.4 provides a detailed description of the reliability tool and the proposed end-of-lifetime analysis.

As future work, the proposed reliability simulation tool should be enhanced to regard other aging mechanisms like HCI or TDDB. Furthermore, the method can be extended to consider also the write stress on the memory array. With the help of this novel tool, different stress balancing techniques can be explored. One example would be to choose an SRAM design, which is split in two arrays with different timing characteristics: Array one is a 1-cycle array, array two is a 2-cycle array, meaning that the time to access the memory requires either one 1 or 2 clock cycles. With this setup it is now possible to distribute only timing critical data on the 1-cycle array to spare the array from exacerbated aging since it experiences less stress. The main part of the workload can then be spread onto the 2-cycle array, since the timing of this array is less critical. Having a split memory array would also enable to exploit recovery effects. If one of the arrays reaches a critical degradation limit, the second array could be utilized, while array one can be set into recovery mode.

Chapter 4.5.1 proposes the Mitigation of AGIng Circuitry (MAGIC), a low-cost circuitry to effectively mitigate aging in Sense Amplifiers (SAs) by wear-leveling. To avoid exacerbated aging in highly-used addresses, MAGIC modifies the mapping of SRAM banks to physical addresses and distributes the stress onto the complete SRAM array. An extensive study for various workloads, temperatures and supply voltages proofs the effectiveness of the mitigation technique. The proposed mitigation scheme MAGIC can mitigate the degradation in the read-path up to 26% for three years of aging while introducing minimal area/performance overhead. An application-aware end-of-life analysis of the SRAM shows that this translates into 3x longer lifetime. Since AppAwareAge is capable of analyzing SRAM architectures of arbitrary size and granularity, Chapter 4.5.2 proposes a second mitigation technique, utilizing the proposed tool as an SRAM design exploration framework (SDE) that generates and characterizes memories of different array granularity (e.g. number of banks/rows/words)

with detailed simulations to find the most reliable configuration in terms of aging for the intended set of applications. Since the array granularity has a notable impact on the aging rates of the memory an exploration of all possible memory configurations early in the design phase can significantly improve memory reliability. The proposed mitigation scheme SDE can mitigate the degradation in the read-path up to 32% for three years of aging while the area overhead is restricted to a new set of SAs for each additional bank.

As future work, MAGIC should be evaluated regarding its effectiveness to mitigate aging in the SRAM cells. Furthermore, both mitigation techniques should be investigated regarding HCI, which could negatively impact the performance improvement of the mitigation technique. Even though the simulation results are expected to be very close to silicon measurement data, an evaluation of the mitigation techniques in silicon will provide further insights to explore their ultimate limits and capabilites. Unfortunately, a silicon evaluation could not be realized during the course of this thesis. Generally, it should be investigated if any of the presented methods and insights are applicable to other memory technologies.

The second part of this work focuses on the reliability assessment during the design phase CPUs and therefore introduces, a real-time power, temperature and aging monitor system (eTAPMon) for FPGA prototypes of MPSoCs in Chapter 5. To develop efficient power management and resource allocation strategies early in the design phase, the hardware monitor system is placed on an FPGA prototyping platform together with a prototype of an MPSoC. Since the evolution of power, temperature and aging on the FPGA is significantly different from the targeted ASIC, a modeling approach was developed that can be used to emulate the behavior of the corresponding ASIC monitors. The models are developed through the characterization of the target ASIC design and are placed inside the hardware monitors. The real-time power monitor approach models the behavior of ASIC power monitors based on an instruction-level energy model. The temperature monitor approach contains a linear regression model obtained from thermal offline simulations. The aging monitor approach is based on a critical path model, which calculates the remaining timing slack. An accelerated aging emulation is possible to predict aged ASIC behavior. The monitor system was implemented on an FPGA board and evaluated for a selected operating scenario for nominal process corners, where it provides useful insights into the power, temperature and aging behavior of the system.

As future work the temperature model should be extended to capture transient temperatures instead of providing steady-state temperatures. Although this was a valid first attempt towards an emulated temperature monitor, the temperature monitor should be improved since this assumption might not be able to capture important heat fluctuations and hence leads to a loss of accuracy. Furthermore, the accuracy of the aging prediction can be improved by incorporating a realistic duty factor. The usefulness of the monitor system should moreover be evaluated for more complex operating scenarios. Especially, the effect of a running operating system needs to be evaluated to develop aging-aware power management strategies to minimize the impact of aging on MPSoCs.

With nowadays systems finding their way into every aspect of our lives, from smartphones to autonomous driving, the demand for highly reliable systems is more important than ever. Thus, accurate prediction and observability of system reliability enables to include necessary safety margins or mitigation schemes right from the beginning of the design phase. Taking necessary countermeasures to guarantee the lifetime of the product can thus considerably contribute to the safety of human lives.

List of Figures

3.1 Development of reliability/variability and transistor costs for the different technology nodes [61] 16
3.2 Failure Rate Curve (Bathtub Curve) 17
3.3 BTI-induced V_{th} shift during stress and recovery 20
3.4 NBTI - stress phase 21
3.5 NBTI - recovery phase 21

4.1 The trend of embedded memory content in high-end SoCs 30
4.2 SRAM Block Architecture [81]. To mitigate aging, the blue box labeled *MAGIC* will be inserted in front of Bank Decoder (See Sec. 4.5.1), the box is not present in the standard SRAM architecture. 32
4.3 Cross-section of the SRAM read-path 33
4.4 6-Transistor SRAM Core Cell 34
4.5 Voltage Latch-based Sense Amplifier 36
4.6 Signal diagram to analyze the SA read time and stress times 37
4.7 Graphical Approach to obtain the SNM using the Butterfly Plot . . 39
4.8 Sensing Delay Measurement 43
4.9 Bit line Swing Measurement 43
4.10 Individual Contribution of the SA to ΔSD and ΔBS over $RSP_{0,k,l}$-$RSP_{1,k,l}$. A positive ΔSD / a negative ΔBS degrades the performance of the SRAM. 47
4.11 Individual Contribution of the SA to ΔSD and ΔBS over $DF_{0,i,j,k}$-$DF_{1,i,j,k}$. A positive ΔSD / a negative ΔBS degrades the performance of the SRAM. 49
4.12 Application-Aware Aging Analysis Flow 51
4.13 Circuit diagram for MAGIC XOR Remapping Block 56
4.14 Bank Remapping Principle with MAGIC 57
4.15 SRAM Design Exploration Framework Flowchart 61

5.1 Projections of Dark Silicon in future Technology Nodes [90] 64

5.2 Generic invasive multi-processor architecture including several loosely-coupled processors (standard RISC CPUs and invasive cores, so-called i-Cores) as well as tightly-coupled processor arrays (TCPAs) 66
5.3 State Chart of an Invasive Program 67
5.4 Implementation of the eTAPMon for a LEON3 quad core design. . 69
5.5 Emulated Power Monitor. 70
5.6 Emulated Temperature Monitor. 72
5.7 Behavior of Duty Cycle in Aging Function. 74
5.8 ΔV_{th}-values monitored from the FPGA Prototype during the Standard Emulation Scenario. 75
5.9 Generation of critical path model for the emulated Aging Monitor. 76
5.10 Accelerated aging 77

6.1 Memory Usage and Worst-case RSP (over all SAs) for different applications 81
6.2 Worst-case ΔSD (over all SAs) for various applications - due to SA aging only and combined cell and SA aging 83
6.3 Worst-case ΔBS (over all SAs) for various applications - due to SA aging only and combined cell and SA aging 83
6.4 ΔSD over Time - due to SA aging only and combined cell and SA aging 85
6.5 ΔBS over Time - due to SA aging only and combined cell and SA aging 86
6.6 ΔSD for different Temperatures - due to SA aging only and combined cell and SA aging 87
6.7 ΔBS for different Temperatures - due to SA aging only and combined cell and SA aging 87
6.8 ΔSD for different Supply Voltages - due to SA aging only and combined cell and SA aging 89
6.9 ΔBS for different Supply Voltages- due to SA aging only and combined cell and SA aging 89
6.10 Worst-case RSP (over all SAs) for different applications with/without MAGIC aging mitigation 91
6.11 Worst-case ΔSD (over all SAs) for various applications - due to combined SA and cell aging with/without MAGIC aging mitigation 91
6.12 Worst-case ΔBS (over all SAs) for various applications - due to combined SA and cell aging with/without MAGIC aging mitigation 92

6.13 ΔSD over Time - due to combined SA and cell aging with/without MAGIC aging mitigation . . . 93
6.14 ΔBS over Time - due to combined SA and cell aging with/without MAGIC aging mitigation . . . 94
6.15 ΔSD for different Temperatures - due to combined SA and cell aging with/without MAGIC aging mitigation . . . 94
6.16 ΔBS for different Temperatures - due to combined SA and cell aging with/without MAGIC aging mitigation . . . 95
6.17 ΔSD for different Supply Voltages - due to combined SA and cell aging with/without MAGIC aging mitigation . . . 96
6.18 ΔBS for different Supply Voltages- due to combined SA and cell aging with/without MAGIC aging mitigation . . . 96
6.19 Lifetime of SRAM for various workloads . . . 98
6.20 Read Addresses of representative Set of Applications . . . 101
6.21 Write Addresses of representative Set of Applications . . . 101
6.22 Worst-case RSP (over all SAs) over all Granularity Configurations 102
6.23 RSP_0, 64 banks, 64 rows and 4 words . . . 103
6.24 RSP_1, 64 banks, 64 rows and 4 words . . . 103
6.25 RSP_0, 2 banks, 512 rows and 16 words . . . 104
6.26 RSP_1, 2 banks, 512 rows and 16 words . . . 104
6.27 Worst-case ΔSD over all Granularity Configurations . . . 106
6.28 Power Monitoring of the Four cores. . . . 108
6.29 Temperature Monitoring of the Four cores. . . . 109
6.30 Slack Monitoring of the Four cores for Nominal Corner. . . . 110
6.31 Slack Monitoring of the Four cores with accelerated Aging. . . . 111

List of Tables

4.1 Modified bank addresses from XOR remapping circuit 57

6.1 Area overhead of MAGIC 99
6.2 Memory Configurations 100
6.3 eTAPMon Hardware Overhead 112

Bibliography

[1] G. E. Moore, "Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff," *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, 2006.

[2] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideout, E. Bassous, and A. R. Leblanc, "Design of ion-implanted MOSFETs with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.

[3] Semiconductor Industry Association, "International technology roadmap for semiconductors," 2009.

[4] F. Kriebel, M. Shafique, S. Rehman, J. Henkel, and S. Garg, "Variability and reliability awareness in the age of dark silicon," *IEEE Design Test*, vol. 33, no. 2, pp. 59–67, 2016.

[5] N. Hardavellas, "The rise and fall of dark silicon," *USENIX ;login*, vol. 37, pp. 7–17, 2012.

[6] J. Henkel and H. Amrouch, "Designing reliable, yet energy-efficient guard-bands," in *IEEE International Conference on Electronics, Circuits and Systems*, pp. 540–543, 2016.

[7] B. Mohammad, *Embedded memory design for multi-core and systems on chip*. Springer, 2014.

[8] M. Wirnshofer, *Variation-aware adaptive voltage scaling for digital CMOS circuits*. Springer Series in Advanced Microelectronics, 2013.

[9] D. Kraak, I. Agbo, M. Taouil, S. Hamdioui, P. Weckx, S. Cosemans, F. Catthoor, and W. Dehaene, "Mitigation of sense amplifier degradation using input switching," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 858–863, 2017.

[10] V. Kleeberger, M. Barke, C. Werner, D. Schmitt-Landsiedel, and U. Schlichtmann, “A compact model for nbti degradation and recovery under use-profile variations and its application to aging analysis of digital integrated circuits,” *Microelectronics Reliability*, vol. 54, no. 6-7, pp. 1083–1089, 2014.

[11] M. B. Taylor, “Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse,” in *Design Automation Conference (DAC)*, pp. 1131–1136, 2012.

[12] Semiconductor Industry Association, “International technology roadmap for semiconductors,” 2011.

[13] J. Teich, J. Henkel, A. Herkersdorf, D. Schmitt-Landsiedel, W. Schröder-Preikschat and G. Snelting, *Invasive computing: An overview.* Multiprocessor System-on-Chip – Hardware Design and Tool Integration, Springer, 2011.

[14] E. Glocker, Q. Chen, U. Schlichtmann, and D. Schmitt-Landsiedel, “Emulation of an ASIC power and temperature monitoring system (eTPMon) for FPGA prototyping,” *Microprocessors and Microsystems*, vol. 50, pp. 90–101, 2017.

[15] S. V. Kumar, K. H. Kim, and S. S. Sapatnekar, “Impact of NBTI on SRAM read stability and design for reliability,” in *7th International Symposium on Quality Electronic Design (ISQED’06)*, pp. 6–218, 2006.

[16] A. Carlson, “Mechanism of increase in SRAM Vmin due to negative-bias temperature instability,” *IEEE Transactions on Device and Materials Reliability*, vol. 7, no. 3, pp. 473–478, 2007.

[17] A. Bansal, R. Rao, J.-J. Kim, S. Zafar, J. H. Stathis, and C.-T. Chuang, “Impacts of NBTI and PBTI on SRAM static/dynamic noise margins and cell failure probability,” *Microelectronics Reliability*, vol. 49, no. 6, pp. 642–649, 2009.

[18] I. Agbo, M. Taouil, S. Hamdioui, P. Weckx, S. Cosemans, and F. Catthoor, “BTI analysis of SRAM write driver,” in *2015 10th International Design Test Symposium*, pp. 100–105, 2015.

[19] H. Yang, S. Yang, W. Hwang, and C. Chuang, “Impacts of NBTI/PBTI on timing control circuits and degradation tolerant design in nanoscale CMOS SRAM,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 6, pp. 1239–1251, 2011.

[20] R. Menchaca and H. Mahmoodi, "Impact of transistor aging effects on sense amplifier reliability in nano-scale CMOS," in *13th International Symposium on Quality Electronic Design*, pp. 342–346, 2012.

[21] I. Agbo, M. Taouil, D. Kraak, S. Hamdioui, H. Kükner, P. Weckx, P. Raghavan, and F. Catthoor, "Integral impact of BTI, PVT variation, and workload on SRAM sense amplifier," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1444–1454, 2017.

[22] I. Agbo, M. Taouil, D. Kraak, S. Hamdioui, P. Weckx, S. Cosemans, F. Catthoor, and W. Dehaene, "Impact and mitigation of SRAM read path aging," *Microelectronics Reliability*, vol. 87, pp. 158–167, 2018.

[23] J. Kinseher, L. Heiß, and I. Polian, "Analyzing the effects of peripheral circuit aging of embedded SRAM architectures," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pp. 852–857, 2017.

[24] A. Gebregiorgis, M. Ebrahimi, S. Kiamehr, F. Oboril, S. Hamdioui, and M. B. Tahoori, "Aging mitigation in memory arrays using self-controlled bit-flipping technique," in *The 20th Asia and South Pacific Design Automation Conference*, pp. 231–236, 2015.

[25] Y. Kunitake, T. Sato, and H. Yasuura, "Signal probability control for relieving NBTI in SRAM cells," in *11th International Symposium on Quality Electronic Design (ISQED)*, pp. 660–666, 2010.

[26] S. Wang, T. Jin, C. Zheng, and G. Duan, "Low power aging-aware register file design by duty cycle balancing," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 546–549, 2012.

[27] T. Siddiqua and S. Gurumurthi, "Recovery boosting: A technique to enhance NBTI recovery in SRAM arrays," in *IEEE Computer Society Annual Symposium on VLSI*, pp. 393–398, 2010.

[28] H. Amrouch, T. Ebi, and J. Henkel, "Stress balancing to mitigate NBTI effects in register files," in *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 1–10, 2013.

[29] S. Kothawade, K. Chakraborty, and S. Roy, "Analysis and mitigation of nbti aging in register file: An end-to-end approach," in *2011 12th International Symposium on Quality Electronic Design*, pp. 1–7, 2011.

[30] M. Shafique, M. U. K. Khan, and J. Henkel, "Content-aware low-power configurable aging mitigation for SRAM memories," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3617–3630, 2016.

[31] M. S. Golanbari, N. Sayed, M. Ebrahimi, M. H. M. Esfahany, S. Kiamehr, and M. B. Tahoori, "Aging-aware coding scheme for memory arrays," in *22nd IEEE European Test Symposium (ETS)*, pp. 1–6, 2017.

[32] S. Wang, G. Duan, C. Zheng, and T. Jin, "Combating NBTI-induced aging in data caches," in *Proceedings of the 23rd ACM International Conference on Great Lakes Symposium on VLSI*, pp. 215–220, 2013.

[33] E. Gunadi, A. A. Sinkar, N. S. Kim, and M. H. Lipasti, "Combating aging with the Colt Duty Cycle Equalizer," in *43rd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 103–114, 2010.

[34] A. Calimera, M. Loghi, E. Macii, and M. Poncino, "Partitioned cache architectures for reduced NBTI-induced aging," in *2011 Design, Automation Test in Europe*, pp. 1–6, 2011.

[35] J. Shin, V. Zyuban, P. Bose, and T. M. Pinkston, "A proactive wearout recovery approach for exploiting microarchitectural redundancy to extend cache SRAM lifetime," in *International Symposium on Computer Architecture*, pp. 353–362, 2008.

[36] R. A. Ashraf, N. Khoshavi, A. Alzahrani, R. F. DeMara, S. Kiamehr, and M. B. Tahoori, "Area-energy tradeoffs of logic wear-leveling for BTI-induced aging," in *Proceedings of the ACM International Conference on Computing Frontiers (CF)*, pp. 37–44, 2016.

[37] M. H. Abu-Rahma, Y. Chen, W. Sy, W. L. Ong, L. Y. Ting, S. S. Yoon, M. Han, and E. Terzioglu, "Characterization of SRAM sense amplifier input offset for yield prediction in 28nm CMOS," in *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–4, 2011.

[38] J. Vollrath, "Signal margin analysis for DRAM sense amplifiers," in *Proceedings First IEEE International Workshop on Electronic Design, Test and Applications*, pp. 123–127, 2002.

[39] J. Coburn, S. Ravi, and A. Raghunathan, "Power emulation: a new paradigm for power estimation," in *Proceedings. 42nd Design Automation Conference*, pp. 700–705, 2005.

[40] A. Bhattacharjee, G. Contreras, and M. Martonosi, "Full-system chip multiprocessor power evaluations using FPGA-based emulation," in *Proceeding of the 13th international symposium on Low power electronics and design (ISLPED)*, pp. 335–340, 2008.

[41] M. E. Ahmad, M. Najem, P. Benoit, G. Sassatelli, and L. Torres, "Adaptive power monitoring for self-aware embedded systems," in *Nordic Circuits and Systems Conference (NORCAS): NORCHIP & International Symposium on System-on-Chip (SoC)*, pp. 1–4, 2015.

[42] N. Ho, P. Kaufmann, and M. Platzner, "A hardware/software infrastructure for performance monitoring on LEON3 multicore platforms," in *24th International Conference on Field Programmable Logic and Applications (FPL)*, 2014.

[43] G. Patrigeon, P. Benoit, and L. Torres, "FPGA-based platform for fast accurate evaluation of ultra low power SoC," in *28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 123–128, 2018.

[44] S. Hesselbarth, T. Baumgart, and H. Blume, "Hardware-assisted power estimation for design-stage processors using fpga emulation," in *24th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 1–8, 2014.

[45] S. Penolazzi, *A system-level framework for energy and performance estimation in system-on-chip architectures*. Information and Communication Technology, KTH Royal Institute of Technology, 2011.

[46] N. Druml, M. Menghin, C. Steger, R. Weiss, A. Genser, H. Bock, and J. Haid, "Emulation-based test and verification of a design's functional, performance, power, and supply voltage behavior," in *21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pp. 328–335, 2013.

[47] J. Haid, C. Bachmann, A. Genser, C. Steger, and R. Weiss, "Power emulation: Methodology and applications for HW/SW power optimization," in *Eighth ACM/IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE)*, pp. 133–138, 2010.

[48] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture," *ACM Transactions on Architecture and Code Optimization*, vol. 1, no. 1, pp. 94–125, 2004.

[49] A. Bartolini, M. Cacciari, A. Tilli, and L. Benini, "Thermal and energy management of high-performance multicores: Distributed and self-calibrating model-predictive controller," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 170–183, 2013.

[50] Z. Wang and S. Ranka, "A simple thermal model for multi-core processors and its application to slack allocation," in *IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*, pp. 1–11, 2010.

[51] D. Atienza, P. G. D. Valle, G. Paci, F. Poletti, L. Benini, G. D. Micheli, J. M. Mendias, and R. Hermida, "Hw-sw emulation framework for temperature-aware design in mpsocs," *ACM Transactions on Design Automation of Electronic Systems*, vol. 12, no. 3, pp. 26–es, 2007.

[52] H. Shen and Q. Qiu, "An FPGA-based distributed computing system with power and thermal management capabilities," in *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, 2011.

[53] F. Beneventi, A. Bartolini, and L. Benini, "On-line thermal emulation: How to speed-up your thermal controller design," in *23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 99–106, 2013.

[54] M. S. Alam and A. Garcia-Ortiz, "An FPGA-based thermal emulation framework for multicore systems," in *27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 1–6, 2017.

[55] D. Lorenz, M. Barke, and U. Schlichtmann, "Efficiently analyzing the impact of aging effects on large integrated circuits," *Microelectronics Reliability*, vol. 52, no. 8, pp. 1546–1552, 2012.

[56] D. Lorenz, M. Barke, and U. Schlichtmann, "Monitoring of aging in integrated circuits by identifying possible critical paths," *Microelectronics Reliability*, vol. 54, no. 6-7, pp. 1075–1082, 2014.

[57] R. Baranowski, F. Firouzi, S. Kiamehr, C. Liu, M. Tahoori, and H. Wunderlich, "On-line prediction of NBTI-induced aging rates," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 589–592, 2015.

[58] A. Birolini, *Reliability Engineering*. Springer Berlin Heidelberg, 2004.

[59] A. Listl, *Modeling and Simulation of NBTI in Digital CMOS Circuits*. Master thesis, Technical University of Munich, Munich, 2013.

[60] N. Pour Aryan, *Monitoring Concepts for Degradation Effects in Digital CMOS Circuits*. Dissertation, Technical University of Munich, Munich, 2015.

[61] T. Austin, V. Bertacco, S. Mahlke, and Y. Cao, "Reliable systems on unreliable fabrics," *IEEE Design Test of Computers*, vol. 25, no. 4, pp. 322–332, 2008.

[62] G. A. Klutke, P. C. Kiessler, and M. A. Wortman, "A critical look at the bathtub curve," *IEEE Transactions on Reliability*, vol. 52, no. 1, pp. 125–129, 2003.

[63] Renesas Electronics, *Semiconductor Reliability Handbook, Rev.2.50*. Renesas Technology Corp., 2017.

[64] M. White and J. Bernstein, *Microelectronics Reliability: Physics-of-Failure Based Modeling and Lifetime Evaluation*. National Aeronautics and Space Administration (NASA), 2008.

[65] S. Minehane, R. Duane, P. O'Sullivan, K. G. McCarthy, and A. Mathewson, "Design for reliability," *Microelectronics Reliability*, vol. 40, no. 8-10, pp. 1285–1294, 2000.

[66] J. Kinseher, *New Methods for Improving Embedded Memory Manufacturing Tests*. Dissertation, Technical University of Munich, Munich, 2017.

[67] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the NBTI effect for reliable design," in *IEEE Custom Integrated Circuits Conference*, pp. 189–192, 2006.

[68] M. A. Alam, "A critical examination of the mechanics of dynamic nbti for pmosfets," in *IEEE International Electron Devices Meeting*, pp. 14.4.1–14.4.4, 2003.

[69] T. Grasser, P. -. Wagner, H. Reisinger, T. Aichinger, G. Pobegen, M. Nelhiebel, and B. Kaczer, "Analytic modeling of the bias temperature instability using capture/emission time maps," in *2011 International Electron Devices Meeting*, pp. 27.4.1–27.4.4, 2011.

[70] S. Kiamehr, F. Firouzi, and M. B. Tahoori, "Aging-aware timing analysis considering combined effects of nbti and pbti," in *International Symposium on Quality Electronic Design (ISQED)*, pp. 53–59, 2013.

[71] S. Zafar, Y. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, and M. Chudzik, "A comparative study of NBTI and PBTI

(charge trapping) in SiO2/HfO2 stacks with FUSI, TiN, Re gates," in *Symposium on VLSI Technology, 2006. Digest of Technical Papers*, pp. 23–25, 2006.

[72] Chenming Hu, "Lucky-electron model of channel hot electron emission," in *International Electron Devices Meeting*, pp. 22–25, 1979.

[73] F. Hsu and S. Tam, "Relationship between MOSFET degradation and hot-electron-induced interface-state generation," *IEEE Electron Device Letters*, vol. 5, no. 2, pp. 50–52, 1984.

[74] D. Lorenz, M. Barke, and U. Schlichtmann, "Aging analysis at gate and macro cell level," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 77–84, 2010.

[75] S. Sahhaf, R. Degraeve, P. J. Roussel, T. Kauerauf, B. Kaczer, and G. Groeseneken, "TDDB reliability prediction based on the statistical analysis of hard breakdown including multiple soft breakdown and wear-out," in *IEEE International Electron Devices Meeting*, pp. 501–504, 2007.

[76] J. C. Cha and S. K. Gupta, "Characterization of granularity and redundancy for SRAMs for optimal yield-per-area," in *IEEE International Conference on Computer Design*, pp. 219–226, 2008.

[77] I. Agbo, M. Taouil, D. Kraak, S. Hamdioui, H. Kükner, P. Weckx, P. Raghavan, and F. Catthoor, "Integral impact of BTI, PVT variation, and workload on SRAM sense amplifier," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1444–1454, 2017.

[78] D. Kraak, M. Taouil, I. Agbo, S. Hamdioui, P. Weckx, S. Cosemans, and F. Catthoor, "Impact and mitigation of sense amplifier aging degradation using realistic workloads," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 12, pp. 3464–3472, 2017.

[79] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *SIGARCH Comput. Archit. News*, vol. 23, no. 1, pp. 20–24, 1995.

[80] R. Saleh, S. Wilton, S. Mirabbasi, A. Hu, M. Greenstreet, G. Lemieux, P. P. Pande, C. Grecu, and A. Ivanov, "System-on-chip: Reuse and integration," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1050–1069, 2006.

[81] A. Pavlov and M. Sachdev, *CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies: Process-Aware SRAM Design and Test*. Springer Publishing Company, 2008.

[82] K. Agarwal and S. Nassif, "Statistical analysis of SRAM cell stability," in *43rd ACM/IEEE Design Automation Conference*, pp. 57–62, 2006.

[83] 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.

[84] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of sram array for yield enhancement in nanoscaled cmos," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, pp. 1859–1880, 2005.

[85] A. Teman, "Dynamic stability and noise margins of sram arrays in nanoscaled technologies," in *IEEE Faible Tension Faible Consommation*, pp. 1–5, 2014.

[86] A. Listl, D. Mueller-Gritschneder, U. Schlichtmann, and S. R. Nassif, "SRAM design exploration with integrated application-aware aging analysis," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1249–1252, 2019.

[87] D. Mueller-Gritschneder, M. Dittrich, J. Weinzierl, E. Cheng, S. Mitra, and U. Schlichtmann, "Etiss-ml: A multi-level instruction set simulator with rtl-level fault injection support for the evaluation of cross-layer resiliency techniques," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 609 –6012, 2018.

[88] M. Jefremow, *Power Efficient and Robust Sense Amplifiers for Embedded Non-Volatile Memories in High-Speed Microcontrollers for Automotive Applications*. Dissertation, Technical University of Munich, Munich, 2014.

[89] A. Listl, D. Mueller-Gritschneder, and U. Schlichtmann, "MAGIC: A wear-leveling circuitry to mitigate aging effects in sense amplifiers of SRAMs," in *IEEE International New Circuits and Systems Conference (NEWCAS)*, pp. 1–4, 2019.

[90] A. M. Rahmani, P. Liljeberg, A. Hemani, A. Jantsch, and H. Tenhunen, *The Dark Side of Silicon: Energy Efficient Computing in the Dark Silicon Era*. Springer International Publishing, 2017.

[91] A. Weichslgartner, S. Wildermann, M. Glaß, and J. Teich, *Invasive Computing for Mapping Parallel Programs to Many-Core Architectures*. Springer Singapore, 2018.

[92] A. Kanduri, A. M. Rahmani, P. Liljeberg, A. Hemani, A. Jantsch, and H. Tenhunen, "A perspective on dark silicon," in *The Dark Side of Silicon: Energy Efficient Computing in the Dark Silicon Era*, pp. 3–20, Springer International Publishing, 2017.

[93] Wei Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," in *Proceedings. 41st Design Automation Conference, 2004.*, pp. 878–883, 2004.

[94] A. Listl, D. Mueller-Gritschneder, F. Kluge, and U. Schlichtmann, "Emulation of an ASIC power, temperature and aging monitor system for FPGA prototyping," in *IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS)*, pp. 220–225, 2018.

[95] Synopsis, *Silicon Smart*. http://www.synopsys.com.

[96] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *IEEE Transactions on Electron Devices*, vol. 53, no. 11, pp. 2816–2823, 2006.

[97] Synopsis, *Prime Time*. http://www.synopsys.com.

[98] OpenRISC, *OpenRISC 1000 Architecture Manual V1.1.* http://opencores.org, 2014.