Marco Ottavi
Dimitris Gizopoulos
Salvatore Pontarelli *Editors*

Dependable Multicore Architectures at Nanoscale

COST
EUROPEAN COOPERATION
IN SCIENCE & TECHNOLOGY

Springer

Dependable Multicore Architectures at Nanoscale

Marco Ottavi · Dimitris Gizopoulos
Salvatore Pontarelli
Editors

Dependable Multicore Architectures at Nanoscale

Springer

Editors

Marco Ottavi
Department of Electronic Engineering
University of Rome Tor Vergata
Rome
Italy

Salvatore Pontarelli
National Inter-University Consortium for Telecommunications (CNIT)
Rome
Italy

Dimitris Gizopoulos
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens
Athens
Greece

ISBN 978-3-319-54421-2 ISBN 978-3-319-54422-9 (eBook)
DOI 10.1007/978-3-319-54422-9

Library of Congress Control Number: 2017943827

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my wife Fausta and my daughter Aurelia.

—Marco Ottavi

To my loves: my wife Georgia, my daughter Despina and my son Nikiforos.

—Dimitris Gizopoulos

To our Ph.D. advisor Adelio Salsano in memory of his dedication to work and his love for life.

—Salvatore Pontarelli and Marco Ottavi

Foreword I

Dependable Multicore Architectures at Nanoscale is the latest and most exciting addition to the technical literature of fault-tolerant computing and dependable computer architecture.

In the nanoscale manufacturing era in which correct circuit operation is jeopardized by multiple and sophisticated sources, or just be plain device unreliability, the understanding and the mitigation of circuits for dependable operation is commonly considered a mandatory design requirement. Dependability enhancement measures and mechanisms are widely utilized today in computing and electronic systems across the entire performance spectrum, from the high complexity of a super-computer down to handheld/embedded devices in mobile systems.

This edited volume captures an in-depth view of the state of the art in this ever-changing research and development domain. The chapters cover a broad treatment of the underlying challenges and related methodologies to deal with the increasing unreliability of state-of-the-art multicore architectures. In addition to capturing application-independent features, this book reports also contributions from major corporations and high-tech companies; the editors have also enlarged the scope of this book by involving researchers from major design companies to present views and expectations for future technical directions.

Dependable Multicore Architectures at Nanoscale serves the dual purposes of a textbook for students and active practitioners; the material here presented can serve as a reference as well as inspiration for advance treatment of theoretical and experimental studies, as found in today's dependable computing systems.

October 2016

Fabrizio Lombardi
Northeastern University, Boston, USA

Foreword II

Dependability of computing systems has been a major design concern since the early days of computing, with the goal of achieving high reliability and availability in the presence of unavoidable hardware and software faults. In recent years, the significance of computing systems' dependability has grown even more due to two major trends. The first is the relentless and continuous spread of computer-based controllers into almost every imaginable product and application, including critical and even life-critical ones. The second trend is the tremendous increase in the complexity of computing systems that now consist of many processing cores implemented using billions of devices. This trend is exacerbated by the advances in the sub-micron technology. These have, on one hand, allowed the vast increase in the number of devices per integrated circuit, but on the other hand, brought about new fault mechanisms and considerable process variations. All this has increased considerably the likelihood of a single device (out of a billion) to become defective during manufacturing or to fail (permanently or intermittently) during the system's lifetime.

This book provides a timely and updated survey of the main issues in the design and use of dependable contemporary multicore systems. It contains a comprehensive survey of the prevalent fault mechanisms in the current nanotechnology and describes in detail the available mitigation techniques to counter the effects of the potential faults.

Two unique features of this book must be highlighted. First, it includes an extensive set of actual application scenarios, illustrating the ways different faults can impact the behavior of these applications, and describing the corresponding mitigation techniques that can be applied. Second, it includes chapters contributed by researchers from three multicore design companies. These chapters provide practical perspectives, straight from the trenches, of the different facets of dependability.

I highly recommend this book to researchers and practitioners in the vital, and continuously evolving, field of dependable computing systems.

October 2016

Israel Koren
University of Massachusetts, Amherst, USA

Foreword III

I'm very pleased to write a foreword for this book which collects many significant and compelling contributions of the most active members of the MEDIAN community. I had the opportunity to work in the MEDIAN network of scientists and researchers and I believe that this book is the perfect way to crystallize and provide to a broader audience the discussions, the knowledge exchange, and the research results in the field of dependable multicore architectures achieved during the years in which the MEDIAN action was running. The book provides a solid background in the field of nanoscale reliability threats and possible mitigation strategies. These topics are discussed at different abstraction levels, without forgetting the interconnections and relationships between the various design layers. This makes this book a perfect tool for a reader who wants to approach the topic of dependability of current and next-generation systems. However, I believe that also experts of the field will appreciate this book for updating their knowledge and as a useful reference for their studies. The choice of describing the main threats and the most used mitigation solutions in specific application scenarios (use cases to the transportation, medical, and space) is one of the most intriguing reasons to read this book. At the same time, the vision provided by several scientists, researchers, and engineers working at the first line of the design of nanoscale systems provides a priceless instrument to know what will be the fundamental future trends in the field of dependable system design. This book can be used as a reference book and part of a reading list in a postgraduate course.

October 2016

Dhiraj Pradhan
University of Bristol, Bristol, UK

Preface

The increasing diversity, density, and complexity of electronic devices is enabling the so-called digital revolution with the evident pervasive presence of electronics in everybody's lives. The benefits of this ubiquitous presence are impressive and widespread: from the constant improvement of productivity in the workplace to the societal impact of a constantly connected and sharing community.

Whatever is the considered scenario, all this incredible progress has been relying on the assumption that the devices can be depended on in their application and for their purposes. The foundations for this assumption are based on a constant work behind the scenes of the technical and scientific community aimed at enhancing the dependability of the devices.

Dependability is a broad term that summarizes several aspects of a system, which typically include availability, reliability, maintainability, safety, and security. All these aspects define whether and how the system will behave according to several requirements which can have different levels of priority based on the specific application. In particular, availability measures the amount of time a system is readily operating, reliability measures the continuity of the correct service, maintainability shows the ability of the system of being repaired and/or modified, safety targets the avoidance of catastrophic consequences in the case of lack of service, and finally security targets the resilience of a systems to threats caused by malicious third parties.

There is an impressive amount of examples from technical literature and from the news about the dreadful consequences of the lack of dependability of an electronic system. Lack of dependability may have several negative consequences spreading from the loss of reputation of a manufacturer to the catastrophic loss of lives. Consider for example the automotive scenario, several cases of massive recalls were caused by issues in the electronic system: famously in one of such cases it was speculated that unintended acceleration causing loss of lives was to be ascribed to faults occurring in the drive-by-wire control modules.

Apart from the loss of lives, the economic impact of unreliable hardware must also be taken into account. Depending on the particular product and application domain, the costs associated to lack of dependability can be extremely high:

consider again the example, in the automotive domain, the costs caused by a massive recall of failing hardware, or in consumer electronics, the economic impact in loss of reputation for the manufacturer. Also, there are specific applications where the costs associated with replacing a failing hardware are just prohibitive: consider as an example the space industry where the replacement of failing hardware is all but impossible unless having to sustain huge costs.

A key step to obtaining dependable systems is at manufacturing. Manufacturing dependable digital electronic devices is a process that takes into account several aspects of the life and use of the designed device. At the manufacturing level, designers must take into account the new challenges introduced by the latest technology trends while at a system level the designer must include suitable approaches to counteract the potential occurrence of events that could lead to a non-dependable behavior.

Dependability and manufacturability are very related: from a temporal point of view, manufacturability deals with the cost-effectiveness of chips during production while dependability deals with the correctness of their operation later in the field. Manufacturability and dependability share common challenges and threats, have common objectives, and utilize common solutions regardless of the employment of chips in systems at the low end of performance and power (low-cost embedded systems or consumer electronics) or the high end of the performance (data centers, cloud computing facilities, or extremely powerful supercomputers).

It should be noted that aspects of manufacturability are very specific to the industrial process such as the cost of manufacturing and its accurate relation to yield. These aspects are not easily available to the academic community and therefore are not the focus of this book; instead, the book content is devoted to the actual physical threats and the mitigation techniques used in general and in particular application domains.

This book stems from the view of MEDIAN, a large network of researchers from academia and industry funded by COST[1] collaborating in the areas of manufacturability and dependability of multicore architectures and their deployment in different computing application domains. In particular, the focus is on multicore architectures.

The shift from increasing core clock frequencies to exploiting parallelism and multicore chip architectures has been the main design drive across all application domains in the electronics and computing industry. The introduction of multicore chips enabled a constant increase in delivered performance otherwise impossible to achieve in single-core designs. Multiple microprocessor cores from different instruction set architectures stay at the epicenter of such chips and are surrounded by memory cores of different technologies, sizes, and functionalities, as well as by

[1] COST is the longest-running European framework supporting transnational cooperation among researchers, engineers, and scholars across Europe and is supported by the EU Framework Programme Horizon 2020.

peripheral controllers, special function cores, analog and mixed-signal cores, reconfigurable cores, etc.

The functionality as well as the complexity of multicore chips is unprecedented. This is the aggregate result of several technologies that emerged and matured together the last few decades: (a) manufacturing process now approaching the 10 nm regime and soon expected to go beyond, (b) sophisticated electronic design automation tools assisting and refining every step of the design process, and (c) new processor architectures across the entire spectrum of performance and power consumption.

The book is structured in three parts.

Part I (Chapter "Manufacturing Threats" to Chapter "Application Scenarios") describes the reliability threats of the latest nanoscale technologies and their modeling at different levels of abstraction of complex multicore systems, and shows the impact of these threats in several safety-critical scenarios.

Part II (Chapter "Manufacturing Solutions" to Chapter "Application-Specific Solutions") illustrates the possible mitigation strategies that can be applied to increase the dependability of complex systems. Also in this part, a specific chapter is dedicated to specific application scenarios, showing the relationship between the mitigation solutions and the characteristic of the environment in which the system will operate.

Part III (Chapter "Variation-Mitigation for Reliable, Dependable and Energy-Efficient Future System Design" to Chapter "Roadmap for On-Board Processing and Data Handling Systems in Space") collects the contributions of experts working in companies and public bodies (ARM, ESA, AMD, STMicroelectronics) providing their view about which are the most important and future trends in the field of design of dependable systems.

A detailed breakdown of contents of the chapters is the following.

Chapter "Manufacturing Threats" gives an overview of the reliability threats of the latest nanoscale generations of CMOS technology designs. First, a discussion on the process variability sources, and on the effect on circuit design and achievable performance is presented. After, the different wear-out physical effects such as Bias Temperature Instability (BTI), Hot Carrier Injection (HCI), Random Telegraph Noise (RTN), and Time-Dependent Dielectric Breakdown (TDDB) are analyzed. Finally, the chapter describes the physical phenomena provoking runtime variability effects such as voltage fluctuation and soft errors.

Chapter "Dependability Threats" provides an overview of fault/error models adopted in methodologies for dependability assessment, analysis, and mitigation, and presents an advanced reliability estimation technique for reliability estimation. Faults are categorized based on their applicability in the various abstraction layers. In particular, specific fault models have been included to take into account modern design trends such as FPGAs and NoCs. Furthermore, the chapter also gives special attention to modeling of aging and wear-out effects that arise during the operational life of the devices, causing either transient, intermittent, or permanent failures. The reliability estimation method is extended with the aim to provide a comprehensive system-level model able to consider multi-component architectures. The chapter ends with an overview of the relevant dependability metrics used in methodologies and techniques targeting dependability problems.

Chapter "Application Scenarios" shows several examples of how the faults occurring in modern technologies impact the system design in domains, such as automotive, railroad and transportation, air and space, and medical, where safety-critical and reliable operations are mandatory requirements. It addresses current practices deployed in these different domains and highlights the risks involved when the effects of the ever scaling technologies and related design techniques on system reliability are not properly taken into consideration. The chapter also discusses the growing interest problem of hardware security, which is a common challenge in all the domains.

Chapter "Manufacturing Solutions" starts the second part of the book, where the design solution to the dependability threats discussed in the first part is presented. This chapter focuses on the threat described in Chapter "Manufacturing Threats" and presents the current available solution to mitigate faults. The presented solutions are applied at different design levels, depending on the specific threat to face and on the targeted dependability level. The chapter shows how to face some threat already during the process manufacturing, in which the used materials and the lithographic process are modified to limit the effects of process variability. Also, techniques based on layout design methodology are introduced. On n higher design level, several circuit level and RTL design level technologies are illustrated to different dependability threats (soft errors, voltage droop).

Chapter "Dependability Solutions" presents an overview of existing dependability solutions for processors and multicore processing systems. First, the existing techniques to protect processor cores both at the hardware and software level are discussed. Then the protection of the different memories that are present in a multicore is reviewed in the second section. Finally, the protection of the interconnections and an overview of specific Network on Chip dependability solutions are covered in the last section.

Chapter "Application-Specific Solutions" examines in detail some mitigation solutions applied in specific critical scenarios. It starts from the consideration presented in Chapter "Application Scenarios" considering a broader variety of application domains and their relation to dependability. The chapter shows how the specific design choices are strictly dependent on the application domain and how the selected solution can be different from each other.

Chapter "Variation-Mitigation for Reliable, Dependable and Energy-Efficient Future System Design" is the first chapter of the third part of the book. The chapter provides the view from two ARM researchers about the major issues related to microprocessor dependability design. The chapter focuses on the issues related to process, voltage, and temperature (PVT) variations and the related mitigation strategies. An overview of the various sources of variation and the traditional approaches for variation-mitigation is presented. Afterward, several promising techniques for variation-mitigation are discussed. In particular, in situ aging monitors, error-resilient techniques, and adaptive clocking techniques are examinated. Furthermore, the chapter provides a detailed analysis of the Razor approach, showing the silicon measurement results from multiple industrial and academic demonstration systems that employ Razor.

Chapter "Design for Test and Test Equipment Roadmap" reports the experience of and the view of two researchers at AMD, another big player deeply involved in the design of next-generation multicore processors. The chapter topic is the resilience proportionality design, an interesting methodology to provide efficient and reliable systems. The chapter observes that chip design companies have to make difficult decisions about the exact dependability level that each product in their portfolio should provide and have few hints on which are the specific request and needs of customers and market segments. Therefore, the tradeoff between design cost, deployment cost, and the dependability level is a critical issue to address. The chapter proposes a resilience proportionality approach able to adapt a design to the dependability needs of a wide range of applications and hardware configurations.

Chapter "Resilience Proportionality—A Paradigm for Efficient and Reliable System Design" presents STMicroelectronics view and a roadmap for Design for test and test equipments. The current and future issues of VLSI test are examinated, highlighting how the exasperated operating conditions (very high temperature, severe mission profiles) and the limited confidence of the various adopted fault models (stuck-at, transition, and bridging) enlarge a progressive gap between the effective adherence of fault models to the actual defects present in IC. The chapter offers some perspective analysis on how these challenges can be faced and hopefully resolved. New synergies between DfT, test equipment, and test methods shall be proposed to highlight cause–effect relations. Special attention shall also be given to the sustainability of the costs of the proposed solution.

Chapter "Roadmap for On-Board Processing and Data Handling Systems in Space" gives the view of two European Space Agency (ESA) scientists about the evolution of on-board processing and data handling systems in the space environment. First, the chapter surveys the state of the art in this field and presents the description of a generic On-Board Computers and Data Systems Architecture. Afterward, the chapter continues identifying the historical path in the design of space system, starting from the old space microprocessors going to the current generation systems (multicore, FPGA, etc.), concluding with the expected future trends.

Dependable and manufacturable computing is a broad and intense research area concentrating major research effort worldwide from the circuit, architecture, and software communities. We believe that the snapshot of the area we deliver with this book reveals the tough challenges, the current directions, as well as the research opportunities in the near future. The forthcoming manufacturing technologies and the requirements of specific application domains will determine the advances in the field and the level of investment the industry will put on it.

We sincerely thank the authors of the book chapters for devoting their time and energy in the corresponding chapters, Springer for publishing the book and COST framework for supporting the networking activities of MEDIAN.

Rome, Italy — Marco Ottavi
Athens, Greece — Dimitris Gizopoulos
Rome, Italy — Salvatore Pontarelli

Funded by the Horizon 2020 Framework Programme of the European Union

This book is based upon work from COST Action IC1103 ("Manufacturable and dependable multicore architectures at nanoscale (MEDIAN)"), supported by COST (European Cooperation in Science and Technology).

COST (European Cooperation in Science and Technology) is a funding agency for research and innovation networks. Our Actions help connect research initiatives across Europe and enable scientists to grow their ideas by sharing them with their peers. This boosts their research, career and innovation.

www.cost.eu

Contents

Part I Challenges

Part II Solutions

Part III State of the Art and Vision

Contributors

Lorena Anghel Grenoble Institute of Technology, Phelma, Grenoble Cedex, France

Davide Appello STMicroelectronics S.R.L, Agrate Brianza, MB, Italy

Cristiana Bolchini Politecnico di Milano, DEIB, Milano (MI), Italy

Shidhartha Das ARM Research, Cambridge, UK

Adrian Evans IROC Technologies, Grenoble, France

Gianluca Furano European Space Technology Centre—ESTEC—European Space Agency, Noordwjik, The Netherlands

Sudhanva Gurumurthi AMD Research, Advanced Micro Devices, Inc, Austin, TX, USA

Said Hamdioui Delft University of Technology, Delft, Netherlands

Viacheslav Izosimov Semcon Sweden AB, Linköping, Sweden; KTH Royal Institute of Technology, Stockholm, Sweden

Ben Kaczer Imec, Louvain, Belgium

Saman Kiamehr Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Juan A. Maestro Universidad Antonio de Nebrija, Madrid, Spain

Hans Manhaeve Ridgetop Europe, Bruges, Belgium

Alessandra Menicucci Faculteit Luchtvaart En Ruimtevaarttechniek, TUDelft, Delft, The Netherlands

Maria K. Michael University of Cyprus, Nicosia, Cyprus

Antonio Miele Politecnico di Milano, DEIB, Milano (MI), Italy

Stelios Neophytou University of Nicosia, Nicosia, Cyprus

Antonis Paschalis University of Athens, Athens, Greece

Salvatore Pontarelli Consorzio Nazionale Interuniversitario Per Le Telecomuni cazioni (CNIT), Rome, Italy

Pedro Reviriego Universidad Antonio de Nebrija, Madrid, Spain

Vilas Sridharan RAS Architecture, Advanced Micro Devices, Inc, Boxborough, MA, USA

Mehdi B. Tahoori Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Part I
Challenges

Manufacturing Threats

Saman Kiamehr, Mehdi B. Tahoori and Lorena Anghel

Abstract This chapter introduces an overview of the main reliability threats of last nanoscale generations of CMOS technology designs. In particular, the chapter focuses on sources of process variability and their impact on circuit design and their performances, but also on the runtime variability such as voltage fluctuations as well soft errors. Further to that we go over the transistor aging provoked by different wear-out physical effects such as Bias Temperature Instability (BTI), Hot Carrier Injection (HCI), Random Telegraph Noise (RTN) and Time-Dependent Dielectric Breakdown (TDDB).

1 Reliability Issues

With aggressive downscaling of CMOS technology into deep nanometer, reliability has become a major issue [1]. In this section, the general sources of reliability issues in current technology nodes are briefly explained.

The sources of unreliability in current technology nodes can be categorized into three different categories: (i) variability issues, (ii) transient faults and soft errors (iii) permanent faults, all of them closely related to the fabrication process and to actual economical and physical difficulties to further improve the fabrication process [2], to the stochastic fluctuations of dopants in transistor channel and the thin oxide thickness [3] and to the intrinsic mechanisms of transistor and interconnect aging [4].

S. Kiamehr · M.B. Tahoori (✉)
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
e-mail: mehdi.tahoori@kit.edu

S. Kiamehr
e-mail: kiamehr@kit.edu

L. Anghel
Grenoble Institute of Technology, Phelma, Grenoble Cedex, France
e-mail: lorena.anghel@imag.fr

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_1

Due to variability, the devices/gates/circuits characteristics are different from the intended designed ones. The variability could be due to "*time-zero*" *variation (process variation)* or *runtime variation* such as voltage and temperature variations. Process variation is a natural device parameter variation which makes the properties of fabricated devices different from that of designed ones. In other words, due to process variation different similarly designed transistors/gates will perform (operate) with parametric differences after fabrication. Due to runtime variation, the transistors/gates properties will change (degrade) during the chip operational lifetime.

Runtime variations are routed in different sources such as voltage variation and temperature variation. The voltage and temperature variations are temporal or spatial according to the place of the transistor/gate and they depend on the workload, frequency and time of operation. Therefore, they cause variation on the properties of different transistors/gates at different locations of the circuit and at different time points during the chip operational lifetime.

Transistor aging is a source of runtime variations caused by different wear-out effects such as *Bias Temperature Instability* (BTI), *Hot Carrier Injection* (HCI) and soft *Time-Dependent Dielectric Breakdown* (soft TDDB), which in turn are dependent on process and runtime variations.

All these effects cause the threshold voltage of the transistors to increase and hence the switching delay of the gates that containing these transistors increases which can eventually lead to parametric timing failures if the delay of the circuit does not meet the timing constraints.

In order to deal with these sources of variation, traditional approaches consist in improving the technology process as much as possible, or to add guard-banding as a common approach at the design level. In the guard-banding approach, a timing margin is added to the designed clock cycle to guarantee the correct operation of the circuit during the operational lifetime. A pessimistic guard-banding leads to a performance loss and optimistic guard-banding results in a low reliability of the chip. Therefore, the required timing margin needs to be accurately predicted. Figure 1 shows the components of the required timing margin for IBM Power7+ processor [5]. As shown in this figure, the main components of the timing margin are uncertainty (e.g. global and local process variation), wearout (transistor aging) and voltage and thermal variations.

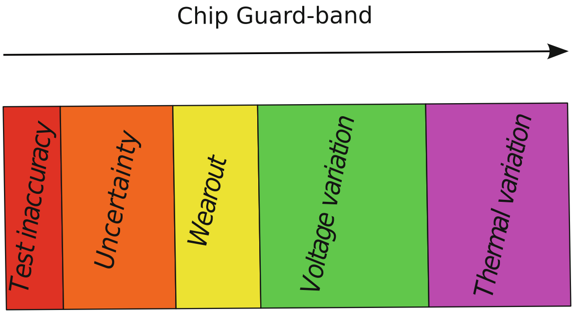


Fig. 1 Components of chip guard-band for the IBM Power7+ [5]

The other category of reliability issues is the transient soft errors caused by alpha particles generated by packaging materials and/or neutrons from cosmic particles. Transient soft errors do not cause a permanent degradation or fault and they lead to a transient computational error [6]. However, since its nature is random, the detection and correction of this type of errors can be very challenging [6]. The soft error can affect memory cell, sequential elements of the circuit but also combinational parts of the circuit. Traditionally, only single errors caused by *single event upsets* were considered as the target of detection and correction methods [7]. However, by continuous scaling of transistor dimensions, the probability that multiple nodes of the circuit are affected simultaneously by a strike [*multi-bit transients in combinational circuits or multi-bit upsets in case of memories* (MBU)] becomes larger which makes the detection and correction even more challenging.

Permanent fault is another important category of reliability issues which has been a concern since early days of electronic industry [6]. *Electromigration* (EM) is one of the most important types of permanent faults which can cause an interconnect disconnection due to the transport of material and it usually happens during runtime. EM is caused by the movement of ions in an interconnect due to the transfer of the momentum between the carriers and the atoms of the interconnect [8]. Permanent faults manifest themselves as logic errors (when properly activated by the circuit inputs) and may provoke catastrophic failures if their correction/decontamination is not handled.

Time-Dependent Dielectric Breakdown (TDDB) is also a major reliability issue which can lead to permanent fault [9]. The material of transistor gate oxide is degraded when a sufficiently high electric field is applied across the gate oxide which leads to an increase of its conductance. In case of a long-term application of electric field a conductive path may be formed through the gate oxide leading to an abrupt increase of gate leakage current. This issue is called hard TDDB and it becomes more severe as the gate oxide thickness becomes thinner due to the technology scaling.

In the following sections, some of the reliability issues which are targeted in this chapter will be explained in more detail.

2 Process Variation

The performance of a circuit is a function of its device characteristic and any variation in the characteristic of devices will lead to a deviation of the circuit performance from its intended designed value. This variation is called process variation and it can cause the circuit to fail if the performance of the circuit does not meet the constraint. Process variation can be categorized into two categories: (i) *Front-end variability* which is the variations caused by manufacturing process of the device (e.g. transistor length variation) and (ii) *Back-end variability* which is the variations caused by manufacturing process of the interconnect [10]. The contribution of these two types of variability is different for various types of reliability concerns (e.g. timing variability or parametric yield) [10]. However, in terms of timing variability, front-end variability is dominant and its contribution in the total path delay is around 90% [11].

2.1 Sources of Front-End Variability

There are different sources of front-end variability, but we will explain the most important issues in the following.

- ***Line Edge Roughness*** **(LER)**: LER is the variation of the edge of the gate along its width which is due to the lithography variations [10] (see Fig. 2) or imperfections during photoresist removal [12]. LER impacts on different device characteristics such as the threshold voltage and the ratio between drive and subthreshold current [10, 13]. Chip manufacturers goal is to reduce this effect by applying corrections, such as optical proximity correction (OPC) or Phase Shift Masks (PSM). However, while shrinking down the gate, it is very challenging to reduce the deviations in the same ratio as the shrinking feature size.
- ***Dielectric thickness variation***: The thickness of the dielectric between the gate and the channel has a large impact on the device characteristic such as the threshold voltage, the drive current and the leakage current [10]. This impact has significantly increased by the continued technology scaling (especially below 30 nm technology process with oxide thickness between 1 and 3 nm) and any variation in the thickness of dielectric will cause important variations in the device characteristic [14].
- ***Random Dopant Fluctuation*** **(RDF)**: The dopant atoms are placed via ion implantation into the channel. The implantation step occurs such that the number and the location of dopant atoms in the channel are random. This phenomenon is called Random Dopant Fluctuation (RDF) which is a statistical variation of the number of implanted dopant and provokes variation of the threshold voltage of the transistor and thus the drive strength. The effect of RDF on the threshold voltage increases by the technology scaling since the number of dopant atoms in the channel decreases with the scaled dimensions [15]. RDF is considered the major source of mismatch for identical adjacent devices [13, 16].

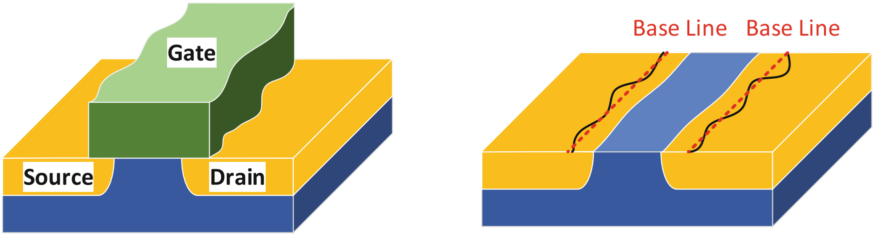


Fig. 2 Line-edge roughness (LER) definition in the transistor

It is common to categorize variations in local and global variations. Global variations (or wafer-to-wafer variation) is mostly due to oxide thickness variations and RDF and affect transistors in the same way. Local variations are random mismatches between neighbour's transistors.

2.2 Process Variation in Emerging Technologies

The aim of new emerging devices such as FinFET technology is to decrease short channel effect as mentioned before in this section. However, they still suffer from some sources of variability. RDF is a major source of threshold voltage variation also for FinFET technology. This is due to the fact the threshold voltage of FinFET devices has a stronger linear dependence on the doping density compared to the conventional MOSFET devices [10]. The other source of threshold voltage variation in this technology is the thickness variation of the silicon fin [10, 17].

2.3 Process Variation Modelling

Since RDF is the major source of variation in advanced MOSFET and FinFET technologies [13, 16] and it significantly affects the threshold voltage of the transistor and the output capacitance of the gate, process variation is considered as the variation of the threshold voltage. A Gaussian (Normal) distribution is considered for the threshold voltage shift of transistors which has a mean value equal to zero and the standard deviation is obtained using Pelgrom model [10, 13, 18]:

$$\mu_{\Delta V_{\mathrm{th}}}^{PV} = 0 \tag{1}$$

$$\sigma_{\Delta V_{\mathrm{th}}}^{PV} = \frac{A_{\Delta V_{\mathrm{th}}}}{\sqrt{WL}} \tag{2}$$

where $A_{\Delta V_{\mathrm{th}}}$ is a technology dependent parameter, W is the width and L is the length of the transistor.

3 Transistor Aging

Transistor aging is one of the major sources of reliability issues in current technologies. The transistor switching delay is degraded over time due to the transistor aging which can eventually cause the circuit to fail if the timing constraint is not met. In this chapter, the focus is on the two major sources of transistor aging which are BTI and HCI. The physical mechanism and modelling of these two effects will be described in more detail in the following sections.

3.1 Bias Temperature Instability (BTI)

BTI is a wear-out phenomenon which gradually degrades the voltage threshold of a transistor and consequently the switching delay of the gate and further to that the circuit path delay. This degradation is monotonous over time. BTI consists of two similar phenomena: (i) *Negative BTI* (NBTI) affecting PMOS transistors and (ii) *Positive BTI* (PBTI) affecting NMOS transistors. NBTI was considered as an important reliability issue for a long time and PBTI was neglected due its small effect on NMOS transistors, however, by the introduction of high-κ metal-gate technologies, PBTI becomes comparable to NBTI [19, 20]. NBTI degradation manifests as a degradation of all electrical parameters of a MOS transistor, under a negative VGS (for PMOS transistor) at relatively high temperatures. It is a static degradation phenomenon, as there is no current in the channel (VDS = 0 V). This degradation gets worse when increasing the temperature, but depends on the type of oxide (SiO_2, SiON, HfO_2, HfSiON) and its thickness [21, 22]. It is usual to quantify this degradation as an important increase of the threshold voltage and a direct current reduction.

In general, there are two main models describing this phenomenon: (i) *Reaction–Diffusion* (RD) model [23–25] and (ii) *Trapping–Detrapping* (TD) model [26, 27]. According to both models, BTI consists of two phases:

- **Stress Phase**: the transistor is under NBTI (PBTI) stress if the gate source of the PMOS (NMOS) transistor is negatively (positively) biased at relatively high temperature. In other words, the transistor is under stress if it is ON. According to the RD model, in this phase, some of the Si–H bonds at the interface of the channel and the gate oxide are broken leading to the generation of interface traps (Reaction). This reaction is triggered by the carriers in the channel (electrons in NMOS and holes in PMOS). The reaction-generated species (hydrogen atoms or molecules) diffuse inside the gate oxide (diffusion) leading to the generation of traps inside the gate oxide. The generation of these traps at the interface of the channel/gate oxide and inside the gate oxide leads to an increase in the threshold voltage of the transistor. The RD mechanism is shown in Fig. 3a. On the other hand, based on the TD model, during the stress phase some pre-existent traps inside the gate oxide capture the charge which leads to an increase in the threshold voltage of the transistor (see Fig. 3b).
- **Recovery phase**: the transistor is in recovery phase if the gate source bias is removed, i.e. when the transistor is OFF. In this phase, according to the RD model, some of the generated traps are removed since some of the generated hydrogen atoms and molecules diffuse back especially for thin oxide thickness structures where the gate tunnel current is important. According to the TD

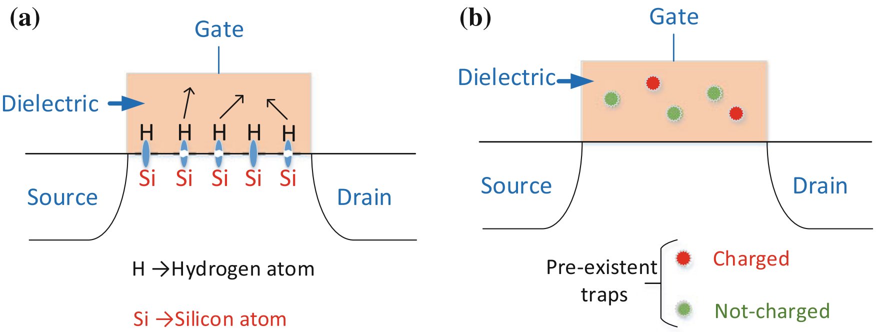


Fig. 3 BTI mechanisms: **a** Reaction–Diffusion (RD) mechanism. **b** Trapping–Detrapping (TD) mechanism

model, during this phase, some of the traps which captured the charge re-emit their charge. In general, the threshold voltage of the transistor decreases during the recovery phase, however, it cannot completely compensate the threshold voltage shift due to the stress phase.

Similar behaviour occurs for NMOS transistor, and we call this phenomenon PBTI. In this case the carriers injected in the gate oxide under a positive VGS are the electrons. PBTI degradation is lower than NBTI, even for most advanced nodes [28, 29]. Next figure shows the evolution of the threshold voltage shift with the time for NMOS and PMOS transistors for 40 nm technology node, for the same input voltage constraint 2.5 V at 125ºC [30].

The difference of behaviour between NBTI and PBTI is explained by the fact that holes are more efficient to generate defects inside the oxide and at the oxide interface than electrons that are injected in the channel by the gate tunnel current that occupy pre-existing defects in the dielectric.

There is still a debate about the model which explains the BTI effect better (TD or RD). According to the literature, although the RD model is suitable to accurately predict the stress phase, it fails to cover the recovery phase [26]. It is observed that even after long time stress (1000 s), threshold voltage drops significantly after 1 s recovery (a very fast recovery) [31]. This fast recovery cannot be explained well by the RD model and it is well explained by TD model [32], however, the RD model is suitable to predict the long-term effect of BTI [26] (Fig. 4).

In previous technology nodes, the BTI effect on transistors was fairly deterministic for a particular workload condition (e.g. temperature and stress) [33]. However, by further downscaling of the transistor dimensions into deca-nanometer range, the number of defects per device decreases leading to a drastic increase in the time-dependent variability of BTI [34]. Thus, it is important to model the stochastic behaviour of BTI in advanced technology nodes. In the following we will explain two BTI models in more detail. One is a deterministic RD model and the other one is a stochastic atomistic trap-based model.

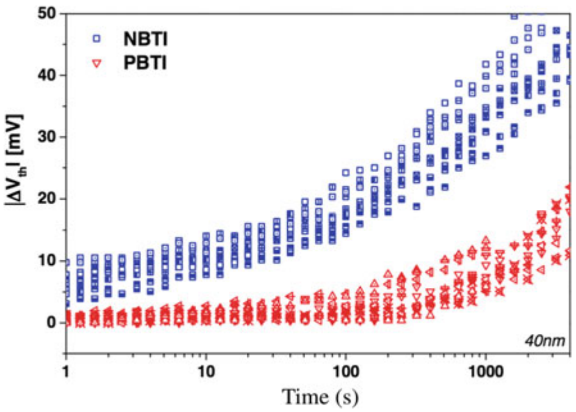


Fig. 4 Threshold voltage shift due to NBTI and PBTI for 40 nm technology node with voltage constraint 2.5 V at 125ºC [30]

3.1.1 Deterministic RD Model

For the deterministic RD model we exploit the model proposed in [23, 24]. The model is proposed for NBTI effect, but since the mechanism of NBTI and PBTI are the same, we have used similar model to address the PBTI effect.

NBTI can be modelled for two different cases: (i) Static NBTI—in which the transistor is under constant stress, and, (ii) Dynamic NBTI—in which the transistor alternatively goes to stress (ON) and recovery (OFF) phases. The static NBTI is more severe compared to the dynamic one since the transistor has no time for recovery in the static NBTI (see Fig. 5a). The threshold voltage shift (ΔV_{th}) due to the static NBTI effect can be expressed by

$$\Delta V_{\text{th}}^{\text{static}} = A\left((1+\delta)t_{\text{ox}} + \sqrt{C(t-t_0)}\right)^{2n} \tag{3}$$

$$A = \left(\frac{qt_{\text{ox}}}{\varepsilon_{\text{ox}}}\right)\sqrt[3]{K^2 C_{\text{ox}}(V_{\text{GS}} - V_{\text{th}})\left(\exp(\frac{E_{\text{ox}}}{E_0})\right)^2} \tag{4}$$

where q is the electron charge, E_{ox} is the electric field of the gate oxide, C_{ox} is the oxide capacitance per area and n is a technology dependent factor which is either equal to 1/4 or 1/6. The other constants and coefficients are summarized in Table 1.

For dynamic NBTI, the ΔV_{th} shift of each stress and recovery phases can be separately expressed by the following equations:

$$\text{Stress:}\, \Delta V_{\text{th}} = \left(K_v(t-t_0)^1/2 + \sqrt[2n]{\Delta V_{\text{th0}}}\right)^{2n} \tag{5}$$

$$\text{Recovery:}\, \Delta V_{\text{th}} = \Delta V_{\text{th0}}\left(1 - \frac{2\xi_1 t_e + \sqrt{\xi_2 C(t-t_0)}}{2t_{\text{ox}} + \sqrt{Ct}}\right) \tag{6}$$

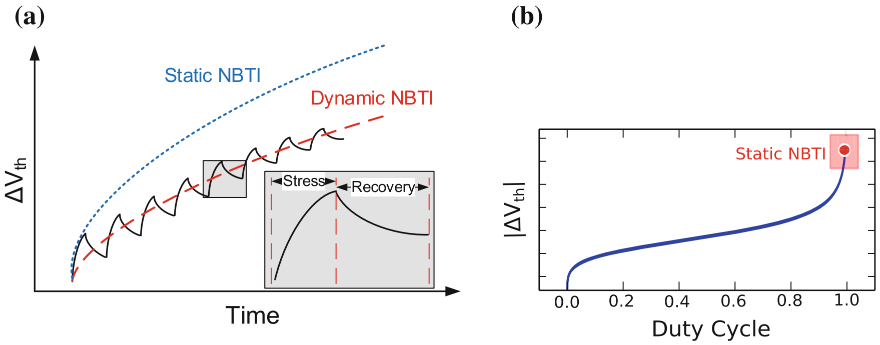


Fig. 5 **a** Static versus dynamic NBTI. **b** The dependency of dynamic NBTI to duty cycle

Table 1 RD model of NBTI-induced ΔV_{th}

NBTI-induced ΔV_{th}		
Static	$A\left((1+\delta)t_{\mathrm{ox}}+\sqrt{C(t-t_0)}\right)^{2n}$	
Dynamic	Stress	$\left(K_v(t-t_0)^1/2+\sqrt[2n]{\Delta V_{\mathrm{th0}}}\right)^{2n}$
	Recovery	$\Delta V_{\mathrm{th0}}\left(1-\frac{2\xi_1 t_e+\sqrt{\xi_2 C(t-t_0)}}{2t_{\mathrm{ox}}+\sqrt{Ct}}\right)$
	Long-term	$\left(\frac{\sqrt{K_v^2\alpha T_{\mathrm{clk}}}}{1-\beta_t^{1/2n}}\right)^{2n}$
Constants and coefficients		
A	$\left(\frac{qt_{\mathrm{ox}}}{\varepsilon_{\mathrm{ox}}}\right)\sqrt[3]{K^2C_{\mathrm{ox}}(V_{\mathrm{GS}}-V_{\mathrm{th}})\left(\exp\left(\frac{E_{\mathrm{ox}}}{E_0}\right)\right)^2}$	
K_v	$\left(\frac{qt_{\mathrm{ox}}}{\varepsilon_{\mathrm{ox}}}\right)^3K^2C_{\mathrm{ox}}(V_{\mathrm{GS}}-V_{\mathrm{th}})\sqrt{C}\exp\left(\frac{2E_{\mathrm{ox}}}{E_0}\right)$	
E_{ox}	$\frac{V_{\mathrm{GS}}-V_{\mathrm{th}}}{t_{\mathrm{ox}}}$	
C	$T_o^{-1}\cdot\exp(-E_a/kT)$	
t_e	if $t-t_0>t_1$ otherwise	t_{ox} $t_{\mathrm{ox}}\sqrt{\frac{t-t_0}{t_1}}-\sqrt{\frac{\xi_2C(t-t_0)}{2\xi_1}}$
$E_a(eV)$	0.49	
$E_0(\mathrm{V/nm})$	0.335	
δ	0.5	
$K(s^{-0.25}\cdot C^{-0.5}\cdot\mathrm{nm}^{-2})$	8×10^4	
ξ_1	0.9	
ξ_2	0.5	
T_o	10^{-8}	

where the constants and coefficients are described in Table 1. Equations 5 and 6 can be exploited to obtain the long-term dynamic NBTI-induced V_{th} shift when transistor undergoes alternate stress and recovery phases:

$$\Delta V_{\mathrm{th}}^{\mathrm{dynamic}} = \left(\frac{\sqrt{K_v^2 \alpha T_{\mathrm{clk}}}}{1 - \beta_t^{1/2n}} \right)^{2n} \tag{7}$$

$$\beta_t = 1 - \frac{2\xi_1 t_e + \sqrt{\xi_2 C(1-\alpha) T_{\mathrm{clk}}}}{2t_{\mathrm{ox}} + \sqrt{Ct}} \tag{8}$$

where T_{clk} is the clock cycle. α in this equation is the *Duty cycle* and defined as the ratio of the time in which transistor is under stress to the total time. NBTI-induced ΔV_{th} is a strong function of the duty cycle as shown in Fig. 5b. The dependence of duty cycle has been confirmed by many measurements performed by different industry teams, on different technology processes [35].

All the equations and related coefficients and constants are summarized in Table 1.

3.1.2 Stochastic Atomistic Trap-Based Model

It is shown that a large portion of the BTI degradation and relaxation during the stress and the recovery phases is due to the charging and discharging of pre-existent gate oxide defects [36]. In previous technology nodes, the BTI effect on transistors was fairly deterministic for a particular workload condition (e.g. temperature and stress) due to the large number of defects in the device (see Fig. 6a). However, by further downscaling of the transistor dimensions into deca-nanometer range, the number of defects per device decreases leading to a drastic increase in the time-dependent variability of BTI [34] (see Fig. 6b). As a result, the lifetime of the device becomes also a *stochastic* value. Figure 6c shows the lifetime of the device for different technology nodes. As shown in this figure, the lifetime spread of smaller devices with lower number of defects is larger.

Therefore it is important to model the intrinsic variation of BTI. In this chapter, we consider the model proposed in [27, 37] for stochastic behaviour of BTI. In this model, each device is characterized by three different factors [27] (see Fig. 7).

- Number of defects (n)
- Defects capture time (τ_{c}): it is defined as the time needed to charge a gate oxide defect during the stress phase.
- Defects emission time (τ_{e}): it is defined as the time needed for the defect to re-emit its charge during the recovery phase.

By knowing these parameters for each device, the total BTI-induced ΔV_{th} of each transistor can be calculated according to Fig. 7b. In this model the total number of defects is obtained from a Poisson distribution:

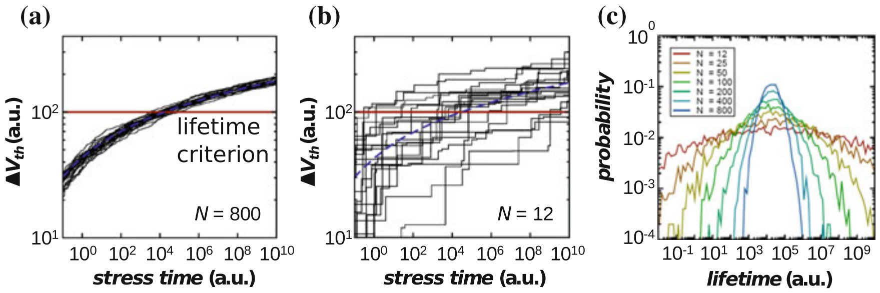


Fig. 6 **a** BTI effect in large devices **b** stochastic behaviour of BTI in deeply scaled devices and **c** lifetime of devices for different technology nodes [37]

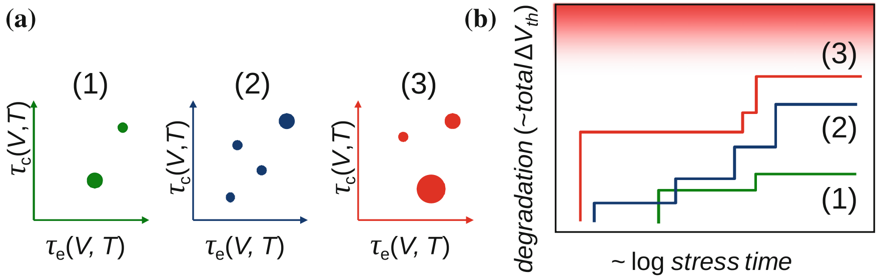


Fig. 7 **a** Parameters affecting BTI for three different devices **b** BTI-induced ΔV_{th} for the three devices [27]

$$n = \text{Poiss}(N_T) \tag{9}$$

$$N_T \propto (L \cdot W) \tag{10}$$

where N_T is the mean number of charged (occupied) defects (traps). L and W are the length and width of the transistor. The effect of each occupied trap is obtained from an exponential distribution:

$$\Delta V_{\text{th}_i} = \text{Exp}(\eta) \tag{11}$$

$$\eta \propto 1/(L \cdot W) \tag{12}$$

where η is the average impact of individual defect on threshold voltage ($\propto$ 1/device area). An analytical description has been derived [27] for the total BTI ΔV_{th} cumulative distribution function as follows:

$$H_{\eta,N_T}(\Delta V_{\text{th}}) = \sum_{n=0}^{\infty} \frac{e^{-N_T} N_T^n}{n!} \left[1 - \frac{n}{n!} \Gamma(n, \frac{\Delta V_{\text{th}}}{\eta}) \right] \tag{13}$$

This formulation allows for an elegant parametrization of the distribution using the average number of defect N_T and the average impact per defect η which further describes the mean and the variance:

$$\mu_{\Delta V_{\mathrm{th}}} = \langle \Delta V_{\mathrm{th}} \rangle = N_T \eta \tag{14}$$

$$\sigma^2_{\Delta V_{\mathrm{th}}} = 2 N_T \eta^2 \tag{15}$$

The average impact per defect η can be extracted from experiments [38]. The average number of defect N_T can be calculated using capture/emission time (CET) maps. CET map describes the probability density function of a broadly distributed defect capture and emission times and it is obtained from experimental data [39, 40] (see Fig. 8a). To build the complete CET map, an analytical two-component bivariate log-normal mixture distribution is used with a probability density of $f_{\mathrm{CET}}(\tau_{\mathrm{c}}, \tau_{\mathrm{e}})$. By integrating the CET map over the entire time domain the total defect density (n_T) and the mean number of available traps in each device $\left(N_T^{\mathrm{avv}}\right)$ can be calculated as follows:

$$n_T = \iint f_{\mathrm{CET}}(\tau_{\mathrm{c}}, \tau_{\mathrm{e}}) d\tau_{\mathrm{c}} d\tau_{\mathrm{e}} \tag{16}$$

$$N_T^{\mathrm{avv}} = W \cdot L \cdot n_T \tag{17}$$

All of these available traps do not contribute on the total BTI-induced V_{th} shift but those which are charged (occupied). The occupancy probability of each trap (P_{occ}) depends on the applied stress waveform (see Fig. 8b) and can be extracted by the following equation:

$$P_{\mathrm{occ}} = \frac{1 - e^{-\frac{\alpha}{f\tau_{\mathrm{c}}}}}{1 - e^{-\frac{1}{f}\left(\frac{\alpha}{\tau_{\mathrm{c}}} + \frac{1-\alpha}{\tau_{\mathrm{e}}}\right)}} \left(1 - e^{-t_{\mathrm{stress}}\left(\frac{\alpha}{\tau_{\mathrm{c}}} + \frac{1-\alpha}{\tau_{\mathrm{e}}}\right)}\right) \tag{18}$$

where α is the duty cycle (the ratio between the stress time to the total time), f is the frequency, and t_{stress} is the total time. Using this occupancy probability (P_{occ}), the CET-active map can be obtained which shows the distribution of active traps (charged defects) according to the corresponding stress waveform (see Fig. 8c). By

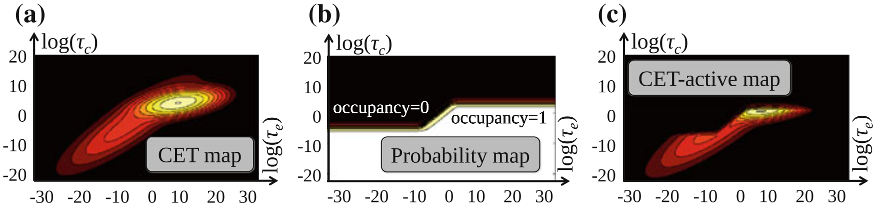


Fig. 8 **a** CET map **b** occupancy probability map **c** CET-active map [34]

integrating the CET-active map over the entire time domain, the average number of defects (N_T) can be obtained by the following equations:

$$\rho = \frac{\iint \mathrm{CET}(\tau_c, \tau_e) P_{\mathrm{occ}}(\tau_{\mathrm{c}}, \tau_{\mathrm{e}}, \alpha, t_{\mathrm{stress}}, f) d\tau_{\mathrm{c}} d\tau_{\mathrm{e}}}{\iint f_{\mathrm{CET}}(\tau_{\mathrm{c}}, \tau_{\mathrm{e}}) d\tau_{\mathrm{c}} d\tau_{\mathrm{e}}} \tag{19}$$

$$N_T = \rho \cdot N_T^{\mathrm{avv}} \tag{20}$$

where N_T is the average number of defects as a result of the applied stress waveform. This parameter is used in Eq. 13 to obtain the CDF of BTI-induced ΔV_{th}.

3.1.3 Process Variation and Stochastic BTI: Are They Correlated?

Since both process variation and stochastic BTI can affect the threshold voltage of a transistor, it is important to consider the correlation of these two effects for the calculation of the total threshold voltage shift of the transistor considering both effects. According to [37, 41], there is no correlation between BTI-induced threshold voltage shift and process variation. However, there is a strong correlation between the standard deviation quantities of threshold voltage shift of these two variation sources since identical sources are responsible for process variation and stochastic BTI variability [42]. From measurements, independently of the technology [42], the correlation has been found to follow the empirical relation:

$$\sigma^2_{\Delta V_{\mathrm{th}}}(t) = \frac{\mu_{\Delta V_{\mathrm{th}}}}{B} \sigma^2_{V_{\mathrm{thPV}}} \tag{21}$$

$$B = 100\,\mathrm{mV} \tag{22}$$

where B is a technology specific parameter. It is important to note that the variances are correlated here, the ΔV_{th} and initial V_{th} are assumed not to be [37, 41].

Assessing the impact of degradation induced time-dependent variability of the V_{th} will be a difficult task in future technologies because of the uncertainty on the BTI critical parameters η and N_T. The correlation between process variation and stochastic BTI, however, gives a powerful predictive method for evaluating existing and future technologies. Combining Eq. (14) and (15) with Eq. (21), η can be directly derived from the initial process variation:

$$\eta = \frac{1}{2B} \sigma^2_{V_{\mathrm{thPV}}} \tag{23}$$

or combining with (Eq. 2)

$$\eta = \frac{A_{\Delta V_{\mathrm{th}}}}{2B\sqrt{WL}} \tag{24}$$

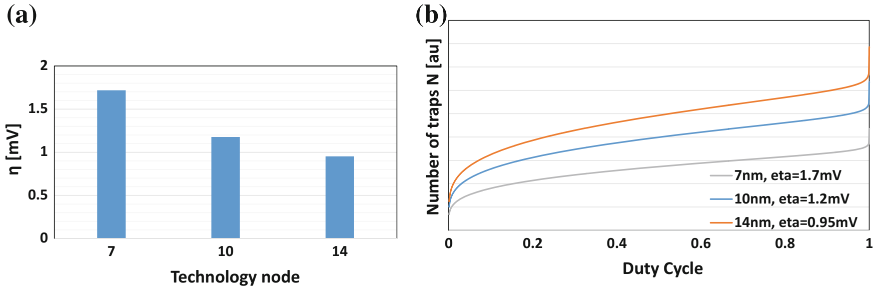


Fig. 9 **a** η calculated using Eq. (24) for different FinFet technologies. **b** Average number of occupied traps as function of DF for different FinFET technologies calculated using Eqs. (18), (19) and (20)

Thus, for simulating future technologies, η is derived directly from the expected Pelgrom's mismatch parameter $A_{\Delta V_{\text{th}}}$ [43–45] and N_T will be calculated using (Eq. 18), (Eq. 19) and (Eq. 20) with a CET map measured on poly silicon oxynitride (SiON) process technology. The scaling of oxide thickness T_{OX} and stress voltage is incorporated by using a power-law extrapolation for the overdrive electric field E_{OX} [46]. Here the V_{th} degradation is proportional to $(E_{\text{OX}})^{\gamma}$, where γ is the voltage acceleration which has a typical value of 3 [47]. Assuming there are no changes in the oxide or oxide quality the extrapolation towards more scaled nodes is done using the following relationship:

$$\frac{\langle \Delta V_{\text{th}_{\text{ref}}} \rangle}{\left(E_{\text{OX,ref}}\right)^{\gamma}} = \frac{\langle \Delta V_{\text{th}_{\text{sim}}} \rangle}{\left(E_{\text{OX,sim}}\right)^{\gamma}} \tag{25}$$

As shown in Fig. 9, values for η and N_T can be readily obtained when using the methodology described above.

3.2 *Hot Carrier Injection (HCI)*

"Hot" carriers are referred to carriers which have a temperature much higher than the lattice temperature. When the transistor is in saturation mode, some of the carriers become "hot" due to the high lateral field and they gain enough energy to overcome channel/gate oxide potential barrier (channel hot carriers) [48]. These channel hot carriers may collide with the silicon atoms in the pinch-off region and generate electron–hole pairs due to the impact ionization. Some of the generated carriers may become "hot" and overcome channel/gate oxide potential barriers [48]. The second type of hot carriers is called avalanche hot carriers.

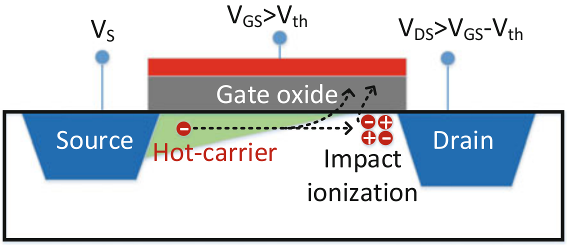


Fig. 10 Hot carrier injection (HCI) physical mechanism

Both channel and avalanche hot carriers may be injected into the gate oxide and damage it generating traps inside the gate oxide or charge existing oxide traps. The gate oxide damage degrades the device characteristic such as the drain current and specially the threshold voltage of the transistor. This phenomenon is called *Hot Carrier Injection* (HCI) or *Channel Hot Carrier* (CHC) which is an important transistor aging issue in nanometer-technology nodes. The physical mechanism of HCI effect is depicted in Fig. 10. HCI describes degradation of the electrical parameters of a MOS transistor under a dynamic stress mode, as it occurs over the whole VDS/VGS range (note that BTI was present only under vertical electrical field with null VDS biasing). We can easily assume that HCI physical phenomenon is worst during the rise and fall bias of the transistors in a given gate.

HCI issue is observed as a critical issue in 80s [48] due to the high lateral electric filed in the technologies used in these period of time. However, from the mid-90s, the supply voltage started to decrease by the technology scaling to decrease the power consumption issue [49]. As a result the lateral electric field decreased and hence, the HCI effect became less by technology scaling. This trend has stopped in recent technology nodes, due to the fact that the supply voltage scaling is slowing down or stopping due to various reasons such as non-scalability of the threshold voltage and the subthreshold slope, signal-to-noise margin issue and process variation. Therefore, the lateral electric field started to increase and hence HCI again has become an important transistor aging issue [49].

HCI mainly affects NMOS transistors and its effect is negligible in PMOS transistors [50] since in the PMOS transistors fewer hot carriers are generated. The reason of this is twofold: (i) shorter mean free path of the holes and (ii) higher oxide barriers for holes.

3.2.1 HCI Model

In this section, we explain the HCI model that is used in the literature. As mentioned previously, the device characteristic such as threshold voltage and subthreshold slope is degraded due to the HCI effect. Here, the model of transistor V_{th} shift as the main effect of HCI is explained (see Fig. 11). Hot carriers are generated

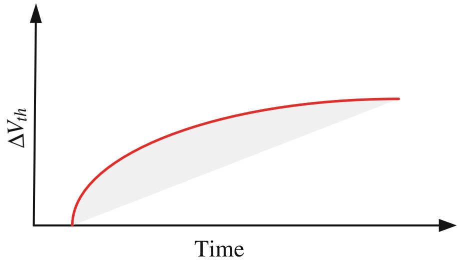


Fig. 11 HCI-induced ΔV_{th} over time

during logic transition and hence the HCI induced V_{th} degradation is a function of switching frequency of the input signal [50, 51]:

$$\Delta V_{\text{th}} = A_{\text{HCI}} \times SW \times f \times e^{\frac{E_{\text{ox}}}{E_1}} \times t^{0.5} \tag{26}$$

$$E_{\text{ox}} = \frac{V_{\text{GS}} - V_{\text{th}}}{t_{\text{ox}}} \tag{27}$$

where A_{HCI} is a technology dependent constant, SW is the switching activity factor, and f is the clock frequency. V_{th} and V_{GS} are the threshold voltage and the gate source voltage of the transistor, respectively. t_{ox} is the oxide thickness, E_1 is a constant equal to $0.8\,\text{V}/\text{nm}$ [52] and t is the total time.

Moreover, it is shown that HCI effect depends on the temperature [49, 53]. Therefore, the HCI model of Eq. 26 is modified as follows:

$$\Delta V_{\text{th}} = A_{\text{HCI}} \times SW \times f \times e^{\frac{-E_a}{kT}} \times e^{\frac{E_{\text{ox}}}{E_1}} \times t^{0.5} \tag{28}$$

where k is the Boltzmann constant and E_a the activation energy for the charge injection into the gate oxide.

3.3 Coupling Models for BTI and HCI Degradations

NBTI and HCI degradation are usually assessed independently one from the other. Their respective degradations are assumed to be additive. However, in [54] it is shown that these two phenomena are interacting and their contributions should be correlated. In fact, as the degradation rate is depending on the damage provoked by carriers, defects created during the two mechanisms are the same, only their respective localizations differ. It is shown through experiments, that the average total (BTI + HCI) degradation is largely overestimated up to a factor of 2 if a simple additive model is used. Thus correlated BTI and HCI models should be used during the evaluation of the degradation for a better accuracy.

3.4 *Random Telegraph Noise (RTN)*

Random Telegraph Noise (RTN) is an important source of runtime variation which is manifested as a low-frequency noise phenomenon and causes a temporal and random fluctuation of transistor electrical parameters, e.g. threshold voltage and drain current [55, 56]. It is shown that RTN is a serious reliability issue for image sensors [57], SRAM [58] and flash memories [59]. Variation due to RTN is increasing with the device downscaling and its effect exceeds process variation in 22 nm technology nodes [60, 61]. Therefore, RTN recently has become a reliability issue also for logic circuits specially for the ones performing under low supply voltage/low power applications [62, 63].

RTN is caused by the stochastic capture/trapping and emission/detrapping of mobile charged carriers into gate dielectric and therefore it shares some common mechanisms with BTI [56, 64, 65]. RTN mechanism and its effect on threshold voltage of transistor is shown in Fig. 12. A carrier in the channel might be captured by a trap in the oxide which leads to an increase in the threshold voltage value of the transistor. The captured carrier will be emitted back after a period of the time and thus the threshold voltage value decreases towards its original value. The capture/emission is a stochastic process and can be described by a two-state Markov chain [55, 66].

The power spectral density of the individual capture/emission process is Lorentzian power spectrum (slope = $1/f^2$), however, the overall RTN effect is the superposition of many capture/emission events which leads to a $1/f$ noise in the frequency domain [56, 67].

3.4.1 RTN Model

RTN has a stochastic behaviour and it is shown that its effect on the circuit leads to a long tail delay distribution [55]. Therefore, it is important to characterize and model the statistical behaviour of RTN. There have been many different models

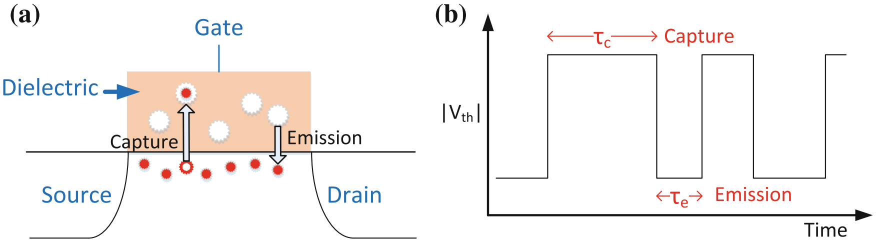


Fig. 12 **a** Capture and emission of mobile carriers **b** threshold voltage fluctuation due to RTN

proposed in the literature; however, here we briefly describe a compact statistical model of the effect of RTN on the threshold voltage of transistor proposed in [68]. In this model, the statistics of number of traps (N_T) and the impact of each single trap on the threshold voltage of transistor $\left(\Delta V_{\text{th}}^{\text{ST}}\right)$ are modelled separately. Then, these two models are combined to obtain a complex model of overall impact of RTN on the threshold voltage of transistor. This model is described in more detail in the following.

Number of Traps (N_T)

It is shown that the number of traps in the transistor follows a Poisson probability distribution [68]:

$$f_T(N_T;\lambda) = \frac{\lambda^{N_T} e^{-\lambda}}{N_T!} \tag{29}$$

where λ is the average number of traps which is a strong function of transistor dimensions and obtained from experimental measurements.

Single-Trap effect on threshold voltage $\left(\Delta V_{\text{th}}^{\text{ST}}\right)$

Single-trap effect on the threshold voltage $\left(\Delta V_{\text{th}}^{\text{ST}}\right)$ has a long tail distribution and therefore it can be modelled by either an exponential distribution [69, 70] or a log-normal distribution [68, 71]. According to [68], a log-normal distribution leads to a better fit to the measured data:

$$f_l\left(\Delta V_{\text{th}}^{\text{ST}}; V_{\text{th0}}, \sigma_l\right) = \frac{e^{-\frac{\left(\ln\Delta V_{\text{th}}^{\text{ST}} - \ln\Delta V_{\text{th0}}\right)^2}{2\sigma_l^2}}}{\sigma_l \Delta V_{\text{th}}^{\text{ST}} \sqrt{2\pi}} \tag{30}$$

$$V_{\text{th0}} = e^{\mu} \tag{31}$$

where σ_l is the log-normal shape parameter and λ is the mean of the distribution of $\ln(\Delta V_{\text{th}}^{\text{ST}})$.

Overall effect of RTN on threshold voltage ΔV_{th}

In order to obtain the overall effect of RTN on threshold voltage (ΔV_{th}), the statistics of N_T and $\Delta V_{\text{th}}^{\text{ST}}$ have to be combined into one comprehensive statistical model. For this purpose, it is assumed that the effects of individual traps on threshold voltage are independent which means that a simple superposition can be used to obtain the overall effects of all traps. Using superposition, the probability distribution function (PDF) of a system with n traps can be expressed as

$$f_{l,n}(\Delta V_{\text{th}}; V_{\text{th0}}, \sigma_l, n) = \int_{-\infty}^{\infty} f_{l,n}(\Delta V_{\text{th}} - u; V_{\text{th0}}, \sigma_l, n-1) \times f_l(u; V_{\text{th0}}, \sigma_l) du \tag{32}$$

Then Eq. 29 can be used to obtain the contribution of the system with n traps into the total RTN effect as

$$a_n = P(N_T = n) = \frac{\lambda^n e^{-\lambda}}{n!} \tag{33}$$

Equations 32 and 33 can be combined to obtain the PDF of overall RTN effect as

$$f_\mathrm{c}(\Delta V_\mathrm{th}; V_\mathrm{th0}, \sigma_l, \lambda) = a_0\delta_0(\Delta V_\mathrm{th}) + \sum_{i=1}^{\infty} a_i f_{l,n}(\Delta V_\mathrm{th}; V_\mathrm{th0}, \sigma_l, i) \tag{34}$$

and cumulative distribution function (CDF) of ΔV_th can be expressed as

$$F_\mathrm{c}(\Delta V_\mathrm{th}; V_\mathrm{th0}, \sigma_l, \lambda) = \int_0^{\Delta V_\mathrm{th}} f_\mathrm{c}(x; V_\mathrm{th0}, \sigma_l, \lambda) dx \tag{35}$$

Concerning RTN, it is worth to note the following issues:

- Physically based 3D TCAD combined with Monte Carlo statistical simulation together with detailed experimental measures and circuit simulation is the present method to better and accurately understand the mechanisms leading to device degradation and the impact on circuit and gates degradations.
- RTN is highly correlated with local process variation, such as the random dopant fluctuation (RDF), line edge roughness (LER), and metal-gate granularity (MGG). Reference [72] shows that the impact on V_th fluctuation is mostly due to RDF and MGG local variations.
- BTI degradation and impact on V_th was compared with RTN impact and it was demonstrated a lack of correlation between these two effects [72].
- RTN has a serious impact on voltage sense amplifiers used for memory designs as the random fluctuation of PMOS drain currents can lead to read errors of stored data [73].
- FDSOI technology, alternatively used for very advanced geometries, shows less local process random variation than bulk technologies, similar RTN-induced variations of current amplitudes and threshold voltages were found [74]. This suggests that in future process nodes operating at lower voltages, RTN will be a major reliability issue.

3.5 Time-Dependent Dielectric Breakdown (TDDB)

Time-Dependent Dielectric Breakdown (TDDB) is an important transistor reliability concern where the quality of gate oxide is degraded over time in presence of high electric fields and it can eventually lead to severe failure in the gate oxide of transistor and a huge leakage current. By technology downscaling, the gate oxide is scaled down; however, the supply voltage does not scale with the same trend. Therefore, the electric field over gate oxide increases and TDDB becomes more of a concern [75].

Generally, there are two types of TDDB called *soft breakdown (SBD)* and *hard break down (HBD)* based on the severity of the problem. In the presence of high electric fields, the channel carriers are trapped inside the oxide dielectric. By increasing the number of traps, they may form a resistive conduction path from gate to channel. In the beginning, the device is still functional; however, this leads to variations in the characteristics of transistors such as threshold voltage and current which is called SBD. The increase in the number of traps will cause the conduction path to become longer which eventually could lead to a catastrophic failure called HBD. When HBD happens, the device is not functional any more since a huge current is drawn from the gate to drain/source of the transistor. Figure 13 shows different phases of oxide breakdown and its impact on leakage path current of transistor [76].

The key reason of causing oxide degradation and eventually breakdown is trap generation. There are different models to explain the trap generation causing TDDB and a correct model is still debatable [9]. Three general models which is well discussed in literature are (1) *Anode Hole Injection* (AHI) model known also as 1/E model [77], (2) *thermo-chemical* model known also as E model [78] and (3) *Anode Hydrogen Release* (AHR) model [79].

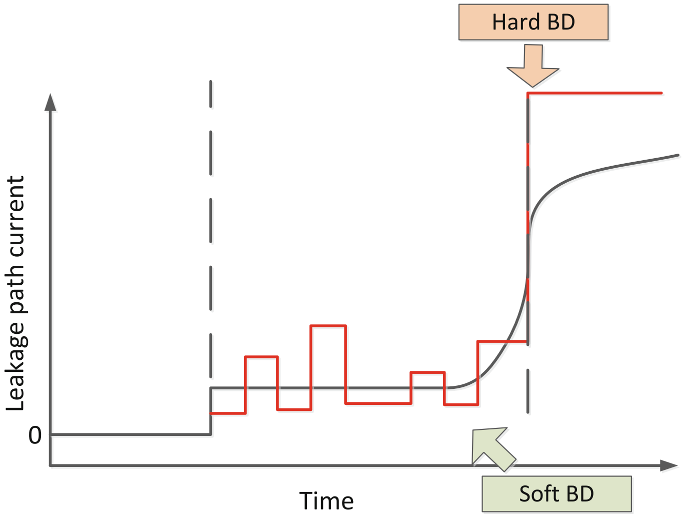


Fig. 13 Time-dependent dielectric breakdown (TTDB) phases [76]

3.5.1 SBD Model

As discussed before, due to SBD the leakage current of transistor increases which might impact on the circuit characteristics such as delay and energy. By technology scaling, the electric field over gate oxide increases which makes the SBD more pronounced [75]. Therefore, it is crucial to model the impact of SBD on the circuit characteristics. A very well-known model for SBD is voltage-dependent power-law gate oxide degradation model [80]. In this model, SBD-induced leakage current increase is modelled by a voltage-dependent current source or a voltage-dependent resistance between the gate and drain/source (see Fig. 14). The voltage-dependent resistance is obtained by the following equation [75]:

$$R_{\mathrm{BD}}(V) = V^{(1-p)}/K \tag{36}$$

where p represents the level oxide degradation and K reflects size of breakdown spot. These two parameters increase over time since the oxide degradation level increases. Extracting the values of K and p is very complicated since the device behaviour after SBD depends on many factors such as technology node, transistor type, oxide area, etc.

3.5.2 HBD Model

Since after HBD the device is not functional any more, HBD normally modelled with a statistical parameter called *time-to-breakdown* (t_{BD}). Generally, t_{BD} is modelled either with Weibull distribution [9] or log-normal distribution [81]. Here, we briefly explain the Weibull distribution model. The CDF of t_{BD} can be described by:

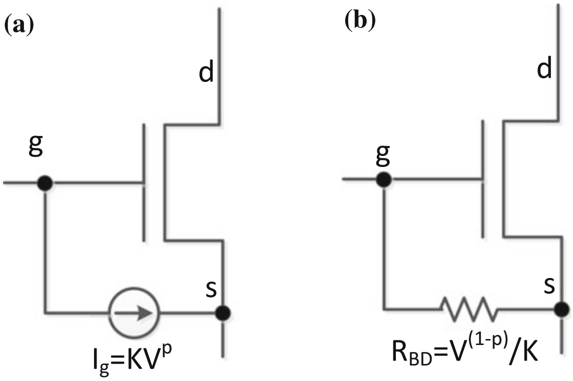


Fig. 14 Power-law SBD model **a** voltage-dependent current source **b** voltage-dependent resistance model [80]

$$F(t) = 1 - \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \quad (37)$$

where β is the shape factor of the distribution and η is the scale factor. Normally, Eq. 37 is rewritten as follows:

$$\ln[-\ln(1 - F(t))] = \beta\ln(t) - \beta\ln(\eta) \quad (38)$$

In which $\ln[-\ln(1 - F(t))]$ can be depicted as a linear function of ln(t) with a slope of β and a y-intercept equal to $-\beta\ln(\eta)$.

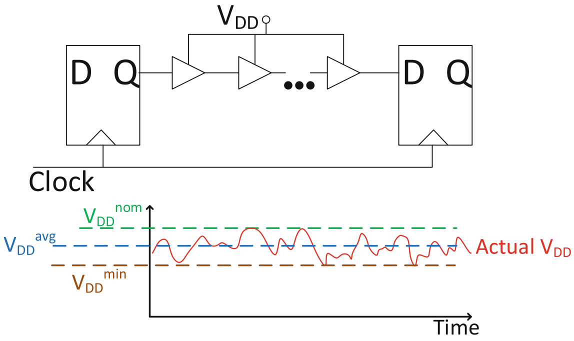
4 Voltage Droop

During workload execution several nodes switch between 0 and 1 and therefore they draw current from power grids. Due to the current drawn from the power grids, the actual supply voltage seen by individual gates inside the circuit decreases and it could vary from time to time and from gate to gate. This phenomenon is called *voltage droop* which is a strong function of the executed workload (see Fig. 15). The voltage droop causes the delay and power dissipation to change and in an extreme case, it may even lead to a functional failure.

The effect of voltage droop on the delay of a simple inverter in 45 nm technology node is shown in Fig. 16. As shown in this figure, a 10% voltage droop can cause the gate delay to increase by more than 20%.

The voltage droop increases by technology scaling since the frequency as well as power densities is increasing [82]. Moreover, by the technology scaling, the sensitivity of the circuit performance to the voltage droop increases and noise margin decreases since the threshold voltage of transistors does not scale down as fast as supply voltage. As a result, the circuit tolerance to the voltage droop is decreasing as the technology geometry scales down [82].

Fig. 15 The supply voltage seen by the gates inside the circuit



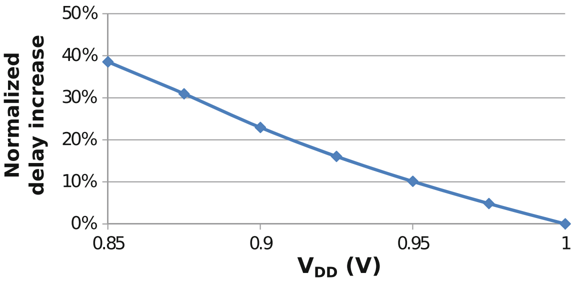


Fig. 16 The delay deviation of a simple inverter versus different supply voltage values in a 45 nm technology node

4.1 Voltage Droop Metrics and Important Parameters

There are two important metrics for voltage droop as shown in Fig. 15:

- **Average voltage droop**: which corresponds to the average supply voltage seen by the gates $(V_{\mathrm{DD}}^{\mathrm{avg}})$. This value correlates with the voltage droop-induced delay degradation of the circuit. It is shown that the effect of voltage droop on timing of a digital path is equal to applying $V_{\mathrm{DD}}^{\mathrm{avg}}$ to the gates of the same path [82]. As a result, this metric needs to be considered to set the timing margin of the design.
- **Maximum voltage droop**: which corresponds to the minimum supply voltage seen by the gates $\left(V_{\mathrm{DD}}^{\min}\right)$. This may cause a failure in the behaviour of the gates or memory cells.

Moreover, there are two components contributing in the total amount of voltage droop:

1. ***IR drop***: which is proportional to the level of the current. *R* represents the resistances of power mesh network, power pads and device package [82].
2. **Ldi/d*t***: which is proportional to the change rate of the current. *L* represents the inductances of the power mesh network, power pads and device packages [82].

It is shown that the contribution of $L\mathrm{d}i/\mathrm{d}t$ is less than that of *IR* drop [82]. Moreover, $L\mathrm{d}i/\mathrm{d}t$ only affects the *maximum voltage droop* and its effect on the *average voltage droop* is negligible [82]. As a result, the contribution of $L\mathrm{d}i/\mathrm{d}t$ on the voltage droop-induced timing degradation of the circuit is small and this parameter can be neglected in the modelling.

4.2 Voltage Droop Model

In this thesis, the power grid is modelled as an R network distributed over the die as shown in Fig. 17. Moreover, for DC analysis, the package model is reduced to a per-connection parasitic resistance [83]. The relationship between voltage (V) and current (*I*) drawn from each node in the power grid can be written as follows [84]:

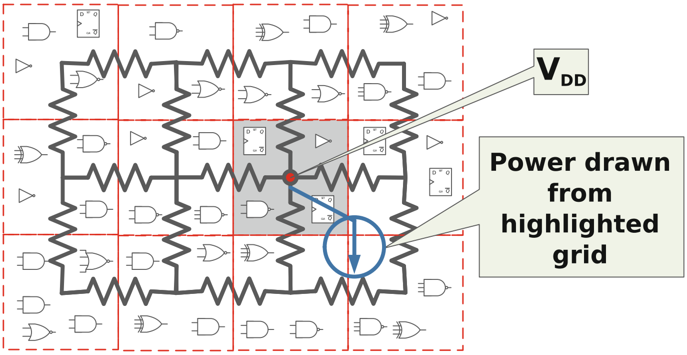


Fig. 17 Equivalent R network model of power grid [83]

$$V = G^{-1}I \tag{39}$$

where G represents the conductance matrix of the power grid. As shown in Fig. 17, current (I) of a grid is calculated by adding the leakage and dynamic power of the gates. The leakage current of each gate is obtained from leakage LUT according to its load capacitance. The switching activity of each node together with parasitic capacitance information obtained from floor-plan is used to estimate the dynamic power of each gate inside the netlist according to the following equation:

$$\text{power}_{\text{dyn}} \sim \text{SW} \cdot C_l \cdot f \cdot V_{\text{DD}}^2 \tag{40}$$

where SW, C_l,f, and V_{DD} are the switching activity, load capacitance of the output node, frequency, and the gate supply voltage, respectively.

There is a negative feedback between voltage droop and power consumption of the circuit such that higher power consumption results in a higher voltage droop which in turn reduces the power consumption. Neglecting this effect might result in a considerable inaccuracy in the estimated voltage droop value.

5 Soft Error

Soft error is a result of the interaction of particles, such as neutron, alpha radiation-induced particles or proton, with device material leading to a perturbation in the device operation. The perturbation can manifest itself as a bit-flip in memory cells or a transient fault in combinational parts of the circuit. This type of error is called "soft" since the device is not permanently damaged and if a new data comes, the device operates correctly again.

Soft error may affect the memory cells and the sequential elements of the circuit by a bit-flip. If only one cell is affected by a single particle, the event is called *Single Event Upset* (SEU). However, a single energetic particle strike can result in upsets in multiple circuit nodes which is called *Multiple Bit Upset* (MBU). By the technology scaling, the dimensions become smaller and devices become closer and hence the MBU rate is increasing [7].

Soft error may also affect the combinational logics. In this case, it manifests itself as a transient pulse (a temporal change in the voltage value of the signal). If the wrong value is stored in the sequential elements, the transient pulse leads to an error. However, the wrong value does not necessarily reach to the sequential element and it may be masked due to different masking phenomena. According to [85], more than 90% of the faults are masked and they do not cause an error. In the following, the most important types of masking are explained in more detail:

- **Electrical masking**: If the transient pulse propagates through successive gates, it may be attenuated such that it cannot propagate more and it is not latched by the sequential element. This phenomenon is called electrical masking and it is a strong function of gates delay. The electrical masking is shown in Fig. 18.
- **Logical masking**: The transient pulse induced by radiation is logically masked if it does not affect the output of the logic due to the functionality of the logical gate. For example if a transient pulse occurs at one input of an AND gate and the other input is logically "0" the output will remains unchanged to value of "0". This phenomenon is depicted in the Fig. 19a.
- **Latching window masking**: This type of masking occurs when the transient pulse reaches to the sequential element outside of its latching window. Figure 19b demonstrates the latching window masking.

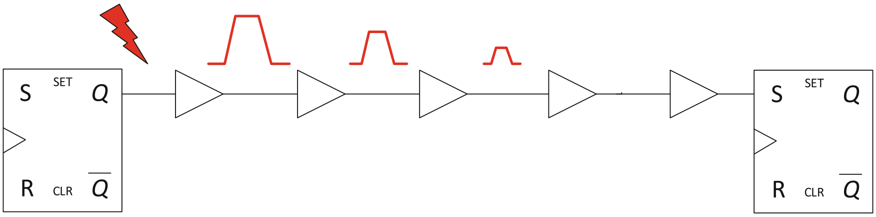


Fig. 18 Electrical masking in logical gates

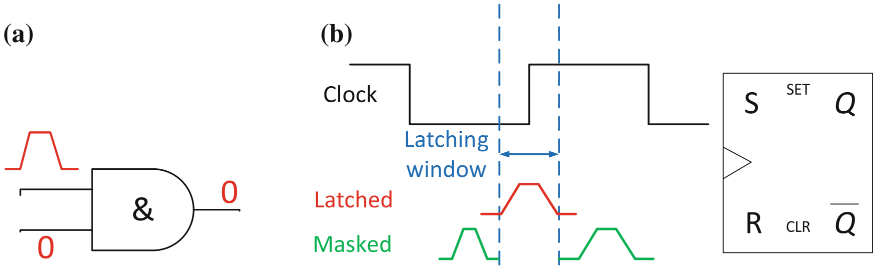


Fig. 19 **a** Logical masking and **b** Latching window masking [86]

5.1 Sources of Radiation

Radiation at ground level causing soft errors comes from different sources. In general, there are two major types of radiation sources:

(1) **Atmospheric radiations**: When a primary cosmic ray (e.g. protons, electrons, photons) enters the atmosphere, it interacts with the molecules of the air leading to the generation of high-energy secondary particles (e.g. neutron, hadrons). The neutron is one of the most important ground level radiation sources affecting circuits, leading to the generation of soft errors. Neutrons are not charged and as a result their interactions with materials do not directly create electron–hole pairs. However, their interactions with material lead to the creation of secondary ionizing particles via "indirect ionization" mechanism. The interaction of generated secondary particles and material in turn leads to the generation of electron–hole pairs. Direct ionization from low-energy protons is another source of soft errors which has become important for technologies beyond 65 nm [87–89]. Proton has a positive electric charge and its mass is slightly lower than neutron.
Muons are also another important part of the atmospheric radiation at ground level. It is a particle similar to electron with a negative charge but with a much greater mass [90]. The probability of muons interaction with material is very small and the interaction happens only for low-energy muons. Like the other charged particles, muons also cause a direct ionization in material. Pions are the other source of atmospheric radiations at ground level. Although they strongly interact with material, their flux density at ground level is very low. Figure 20a shows the spectrum of atmospheric radiations at ground level [91].

(2) **Terrestrial radiations**: Alpha particles are the only terrestrial radiations which cause soft errors in current technologies. An alpha particle consists of two protons and two neutrons (identical to the helium nuclei). ^{238}U, ^{235} U, ^{232} Th are the main sources which emit alpha particles with an energy range of less than 10 meV (see Fig. 20b).

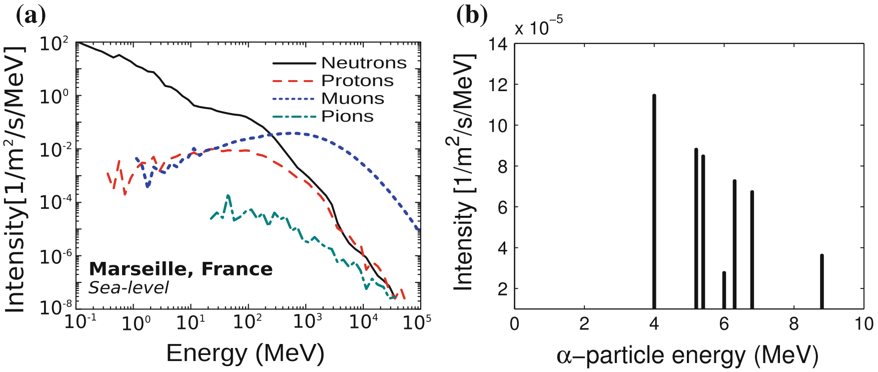


Fig. 20 **a** Neutron, proton, muons, and pions spectrum at sea level [91, 92], **b** alpha particle spectrum assuming that the emission rate is equal to 0.001 $\alpha/\mathrm{h\,cm}^2$ [93]

5.2 Basic Physical Mechanism of Soft Error

In this section, we briefly describe the physical mechanism of soft error. As mentioned before, there are different types of particles affecting the VLSI devices. These particles can directly lead to ionization of materials if the particle is charged (such as protons and alpha particle). In case of neutron, since the particle is neutral, it cannot directly deposit charge on the material. However, the interaction of neutron and material can lead to the generation of charged particles (secondary particles) which in turn leads to the ionization of the material. In the following, the physical mechanism of soft error caused by charged particles is described.

In general, the passage of a charged particle through the device material can be explained in three different steps [91] which are shown in Fig. 21.

- **Charge deposition**: When a charged particle strikes the device, it transmits a large amount of energy to the materials mainly due to inelastic interaction [91]. The deposited energy leads to a generation of electron–hole pairs along the particle path (see Fig. 21a). The energy deposited in the material depends on the particle energy and the material. Moreover, the energy needed to generate the electron–hole pairs depends on the material band-gap. For example, the energy required for generation each electron–hole pair in silicon material is about 3.6 eV. Putting all together, the number of generated electron–hole pairs strongly depends on the particle energy and the material struck.
- **Charge transport**: when the electron–hole pairs are generated due the interaction of particle and material, the generated carriers are transported due to two main mechanisms (see Fig. 21b, c):

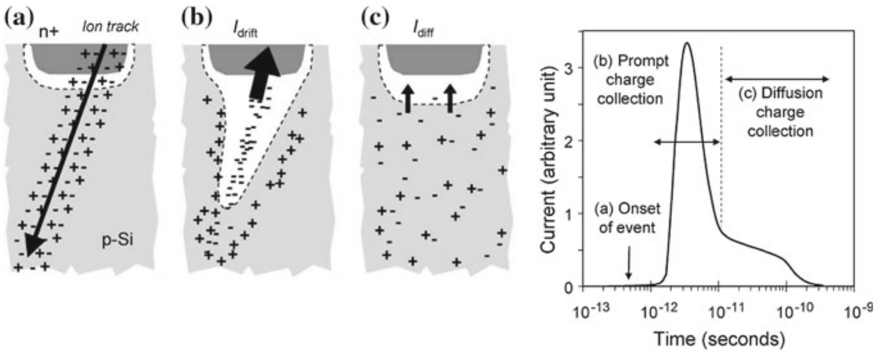


Fig. 21 Physical mechanism of soft error caused by the passage of a charged particle [94]

1. Drift: If the generated carriers are in the regions with an electric filed (e.g. channel), the carriers are transported due to the drift mechanism.
2. Diffusion: If the generated carriers are in the neutral regions, the carriers are transported due to diffusion mechanism from the places with high density of carriers towards the region with less density.

- **Charge collection**: The deposited charge can be collected by the sensitive regions (reversely biased p–n junction due to their strong electric field) and generates a transient current. The generated transient current can lead to a bit-flip in case an SRAM cell or a latch or flip-flop is affected by radiation. In the case of logical gates, the generated transient fault can be propagated through the gates and if it is latched with sequential element, it leads to an error.

6 Summary

In summary, in this chapter, an overview of the most important reliability issues in current technology nodes and their corresponding models are introduced. The focus of the chapter was process variation, transistor aging and its mechanisms such as BTI and HCI,voltage droop and soft error.

References

1. International technology roadmap of semiconductors (itrs). http://www.itrs.net
2. R. Doering, Y. Nishi, Limits of integrated-circuit manufacturing. Proc. IEEE **89**(3), 375–393 (2001)
3. A.J. Bhavnagarwala, X. Tang, J.D Meindl, The impact of intrinsic device fluctuations on CMOS SRAM cell stability. IEEE J. Solid-State Circ. **36**(4), 658–665 (2001)

4. V. Huard, E. Pion, F. Cacho, D. Croain, V. Robert, R. Delater, P. Mergault, S. Engels, L. Anghel, N. Ruiz Amador. A predictive bottom-up hierarchical approach to digital system reliability. In *IEEE International Reliability Physics Symposium (IRPS'12)*, (IEEE Computer Society, 2012), pp 4B–1
5. S. Taylor et al., Power7+: IBM's next generation POWER microprocessor. In *Hot Chips*, vol 24, 2012
6. M.A. Alam, K. Roy, C. Augustine, Reliability-and process-variation aware design of integrated circuits—a broader perspective. In *Reliability Physics Symposium (IRPS), 2011 IEEE International, IEEE*, 2011, pp. 4A–1
7. S. Mitra, K. Brelsford, Y.M. Kim, H.-H. Lee, Y. Li, Robust system design to overcome CMOS reliability challenges. IEEE J. Emerg. Sel. Top. Circ. Syst. **1**(1), 30–41, 2011
8. R. Reis, Y. Cao, G. Wirth, *Circuit Design for Reliability*. (Springer, 2014)
9. J.B. Bernstein, M. Gurfinkel, L. Xiaojun, J. Walters, Y. Shapira, M. Talmor, Electronic circuit reliability modeling. Microelectron. Reliab. **46**(12), 1957–1979 (2006)
10. M. Orshansky, S. Nassif, D. Boning, *Design for Manufacturability and Statistical Design: A Constructive Approach*. (Springer, 2007)
11. M. Orshansky, C. Spanos, C. Hu, Circuit performance variability decomposition. In *Statistical Metrology, 1999. IWSM. 1999 4th International Workshop on* IEEE, 1999, pp. 10–13
12. A. Asenov, S. Kaya, A.R. Brown, Intrinsic parameter fluctuations in decananometer MOSFETs introduced by gate line edge roughness. *IEEE Trans. Electron Devices*, **50**(5), 1254–1260 (2003)
13. K. Kuhn, C. Kenyon, A. Kornfeld, M. Liu, A. Maheshwari, W.-K. Shih, S. Sivakumar, G. Taylor, P. Van Der Voorn, K. Zawadzki, Managing process variation in Intels 45 nm CMOS technology. Intel Techno. J. **12**(2), 2008
14. M. Koh, W. Mizubayashi, K. Iwamoto, H. Murakami, T. Ono, M. Tsuno, T. Mihara, K. Shibahara, S. Miyazaki, M. Hirose, Limit of gate oxide thickness scaling in MOSFETs due to apparent threshold voltage fluctuation induced by tunnel leakage current. IEEE Trans. Electron Devices **48**(2), 259–264 (2001)
15. H. Mahmoodi, S. Mukhopadhyay, K. Roy, Estimation of delay variations due to random-dopant fluctuations in nanoscale CMOS circuits. IEEE J Solid-State Circ **40**(9), 1787–1796 (2005)
16. K.J. Kuhn, Reducing variation in advanced logic technologies: approaches to process and design for manufacturability of nanoscale CMOS. In IEEE International Electron Devices Meeting, 2007. IEDM 2007, IEEE, 2007, pp. 471–474
17. S. Xiong, J. Bokor, Sensitivity of double-gate and FinFETDevices to process variations. IEEE Trans. Electron Devices **50**(11), 2255–2261 (2003)
18. M.J.M. Pelgrom, A.C.J. Duinmaijer, A.P.G. Welbers et al., Matching properties of MOS transistors. IEEE J. Solid-State Circ. **24**(5), 1433–1439 (1989)
19. S. Zafar, Y.H. Kim, V. Narayanan, C. Cabral, V. Paruchuri, B. Doris, J. Stathis, A. Callegari, M. Chudzik, A comparative study of NBTI and PBTI (charge trapping) in Sio_2/HfO_2 stacks with FUSI, TiN, Re gates. In VLSI Technology, 2006. Digest of Technical Papers. IEEE
20. J.S. Stathis, M. Wang, K. Zhao, Reliability of advanced high-k/metal-gate n-FET devices. Microelectron. Reliab. **50**(9), 1199–1202 (2010)
21. M. Denais, C. Parthasarathy, G. Ribes, Y. Rey-Tauriac, N. Revil, A. Bravaix, V. Huard, F. Perrier, on-the-fly characterization of NBTI in ultra-thin gate oxide PMOSFET's. In IEEE International Electron Devices Meeting, 2004. IEDM Technical Digest, IEEE, 2004, pp. 109–112
22. V. Huard, C.R. Parthasarathy, A. Bravaix, T. Hugel, C. Guérin, E. Vincent, Design-in-reliability approach for nbti and hot-carrier degradations in advanced nodes. IEEE Trans. Device Mater. Reliab. **4**(7), 558–570 (2007)
23. W. Wang, V. Reddy, A.T. Krishnan, R. Vattikonda, S. Krishnan, Y. Cao (2007) Compact modeling and simulation of circuit reliability for 65-nm cmos technology. IEEE Trans. Device Mater. Reliab. **7**(4), 509–517, 2007

24. S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, S.A.V.S. Vrudhula. Predictive modeling of the NBTI effect for reliable design. In Custom Integrated Circuits Conference IEEE, 2006. CICC'06. IEEE, 2006, pp. 189–192
25. T. Naphade, N. Goel, P.R. Nair, S. Mahapatra, investigation of stochastic implementation of reaction diffusion (rd) models for NBTI related interface trap generation. In *Reliability Physics Symposium (IRPS), 2013 IEEE International IEEE*, 2013, pp. XT–5
26. V. Huard, C. Parthasarathy, C. Guerin, T. Valentin, E. Pion, M. Mammasse, N. Planes, L. Camus, Nbti Degradation: from transistor to SRAM arrays. In *Reliability Physics Symposium, 2008. IRPS 2008. IEEE International, IEEE*, 2008, pp. 289–300
27. B. Kaczer, T. Grasser, P.J. Roussel, J. Franco, R. Degraeve, L-A. Ragnarsson, E. Simoen, G. Groeseneken, H. Reisinger, origin of NBTI variability in deeply scaled pFETs. In *IEEE Reliability Physics Symposium (IRPS), 2010 IEEE International*, 2010, pp. 26–32
28. M. Wang, R. Muralidhar, J.H. Stathis, B. Paul Linder, H. Jagannathan, J. Faltermeier. Superior PBTI reliability for SOI FinFET technologies and its physical understanding. IEEE Electron Device Lett. **34**(7), 837–839 (2013)
29. K. Taek Lee, W. Kang, E-A. Chung, G. Kim, H. Shim, H. Lee, H. Kim, M. Choe, N-I. Lee, A. Patel et al., Technology scaling on high-K & metal-gate FinFET BTI reliability. In *IEEE Reliability Physics Symposium (IRPS), 2013 IEEE International*, 2013, pp. 2D–1
30. M. Salvia, (2015) *Dedicated Circuits for study Aging induced mecanisms in advanced CMSO technologies, Design and Measures*. Ph.D. Thesis, 2015
31. C. Shen, M-F. Li, C.E. Foo, T. Yang, D.M. Huang, A. Yap, G.S. Samudra, Y.C. Yeo, Characterization and physical origin of fast vth transient in NBTI of pMOSFETs with SiON dielectric. In IEEE International Electron Devices Meeting, 2006. IEDM'06, 2006, pp. 1–4
32. T. Grasser, B. Kaczer, W. Goes, H. Reisinger, T. Aichinger, P. Hehenberger, P.-J. Wagner, F. Schanovsky, J. Franco, M.T. Luque et al., The paradigm shift in understanding the bias temperature instability: from reaction–diffusion to switching oxide traps. IEEE Trans. Electron Devices **58**(11), 3652–3666 (2011)
33. V. Reddy, J.M. Carulli, A.T. Krishnan, W. Bosch, B. Burgess, Impact of negative bias temperature instability on product parametric drift. In International Tset Conference (ITC), Citeseer, 2004, pp. 148–155
34. P. Weckx, B. Kaczer, M. Toledano-Luque, T. Grasser, P.J. Roussel, H. Kukner, P. Raghavan, F. Catthoor, G. Groeseneken, Defect-based methodology for workload-dependent circuit lifetime projections-application to sram. In *IEEE Reliability Physics Symposium (IRPS), 2013 IEEE International*, 2013, pp. 3A–4
35. S. Mahapatra, V. Huard, A. Kerber, V. Reddy, S. Kalpat, A. Haggag, Universality of NBTI-from devices to circuits and products. In *IEEE Reliability Physics Symposium, 2014 IEEE International*, 2014, pp. 3B–1
36. T. Grasser, B. Kaczer, W. Goes, H. Reisinger, T. Aichinger, P. Hehenberger, P-J. Wagner, F. Schanovsky, J. Franco, P. Roussel et al., Recent advances in understanding the bias temperature instability. In IEEE Electron Devices Meeting (IEDM), 2010 IEEE International, 2010, pp. 4–4
37. B. Kaczer, S. Mahato, V. Valduga de Almeida Camargo, M. Toledano-Luque, P.J. Roussel, T. Grasser, F. Catthoor, P. Dobrovolny, P. Zuber, G. Wirth et al., Atomistic approach to variability of bias-temperature instability in circuit simulations. In *IEEE Reliability Physics Symposium (IRPS), 2011 IEEE International*, 2011, pp. XT–3
38. J. Franco, B. Kaczer, M. Toledano-Luque, P.J. Roussel, Jerome Mitard, L-A Ragnarsson, L. Witters, T. Chiarella, M. Togo, N. Horiguchi et al., Impact of single charged gate oxide defects on the performance and scaling of nanoscaled fets. In *IEEE Reliability Physics Symposium (IRPS), 2012 IEEE International*, pp. 5A–4, 2012
39. H. Reisinger, T. Grasser, W. Gustin, C. Schlunder, The statistical analysis of individual defects constituting NBTI and its implications for modeling DC-and AC-stress. In *IEEE Reliability Physics Symposium (IRPS), 2010 IEEE International*, 2010, pp. 7–15

40. T. Grasser, P-J. Wagner, H. Reisinger, T. Aichinger, G. Pobegen, M. Nelhiebel, B. Kaczer, Analytic modeling of the bias temperature Instability using capture/emission time maps. In IEEE Electron Devices Meeting (IEDM), 2011 IEEE International, 2011, pp. 27–4
41. D. Angot, V. Huard, L. Rahhal, A. Cros, X. Federspiel, A. Bajolet, Y. Carminati, M. Saliva, E. Pion, F. Cacho et al., BTI variability fundamental understandings and impact on digital logic by the use of extensive dataset. In *Proceeding of IEEE International Electron Devices Meeting (IEDM)*, 2013, pp. 15–4
42. M. Toledano-Luque, B. Kaczer, J. Franco, P.J. Roussel, M. Bina, T. Grasser, M. Cho, P. Weckx, and G. Groeseneken. Degradation of time dependent variability due to interface state generation. In *2013 Symposium on VLSI Technology (VLSIT),* 2013, pp. T190–T191
43. T. Matsukawa, Y. Liu, W. Mizubayashi, J. Tsukada, H. Yamauchi, K. Endo, Y. Ishikawa, S. O'uchi, H. Ota, S. Migita, Y. Morita, M. Masahara, Suppressing Vt and Gm variability of FinFETs using amorphous metal gates for 14 nm and beyond. In *2012 IEEE International Electron Devices Meeting (IEDM),* 2012, pp. 8.2.1–8.2.4
44. A. Veloso, G. Boccardi, L.-A. Ragnarsson, Y. Higuchi, J.W. Lee, E. Simoen, P.J. Roussel, M.J. Cho, S.A. Chew, T. Schram, H. Dekkers, A. Van Ammel, T. Witters, S. Brus, A. Dangol, V. Paraschiv, E. Vecchio, X. Shi, F. Sebaai, K. Kellens, N. Heylen, K. Devriendt, O. Richard, H. Bender, T. Chiarella, H. Arimura, A. Thean, and N. Horiguchi. Highly scalable effective work function engineering approach for multi-V T modulation of planar and FinFET-based RMG high-K last devices for (sub-)22 nm nodes. In *2013 Symposium on VLSI Technology (VLSIT),* 2013, pp. T194–T195
45. X. Yuan, T. Shimizu, U. Mahalingam, J.S. Brown, K.Z. Habib, D.G. Tekleab, Tai-Chi Su, S. Satadru, C.M. Olsen, Hyun-Woo Lee, Li-Hong Pan, T.B. Hook, J-P. Han, J-E. Park, M-H. Na, K. Rim, Transistor mismatch properties in deep-submicrometer CMOS technologies. IEEE Trans. Electron Devices **58**(2), 335–342 (2011)
46. M. Cho, J.-D. Lee, M. Aoulaiche, B. Kaczer, P. Roussel, T. Kauerauf, R. Degraeve, J. Franco, L. Ragnarsson, G. Groeseneken, Insight into N/PBTI mechanisms in sub-1-nm-EOT devices. IEEE Trans. Electron Devices **59**(8), 2042–2048 (2012)
47. J. Franco, B. Kaczer, P.J. Roussel, J. Mitard, S. Sioncke, L. Witters, H. Mertens, T. Grasser, G. Groeseneken, Understanding the suppressed charge trapping in relaxed- and strained-Ge/SiO_2/HfO_2 pMOSFETs and implications for the screening of alternative high-mobility substrate/dielectric CMOS gate stacks. In *2013 IEEE International Electron Devices Meeting (IEDM)*, 2013, pp. 15.2.1–15.2.4
48. K.-L. Chen, S.A. Saller, I.A. Groves, D.B. Scott, Reliability effects on MOS transistors due to hot-carrier injection. IEEE J. Solid-State Circ. **20**(1), 306–313 (1985)
49. A. Bravaix, C. Guerin, V. Huard, D. Roy, J.-M. Roux, E. Vincent, Hot-carrier acceleration factors for low power management in DC–AC stressed 40 nm NMOS node at high temperature. In *IEEE Reliability Physics Symposium, 2009 IEEE International,* pp. 531–548. 2009
50. A. Tiwari, J. Torrellas, Facelift: Hiding and slowing down aging in multicores. In *Microarchitecture, IEEE/ACM International Symposium*, 2008, pp. 129–140
51. E. Takeda, C.Y.-W. Yang, A. Miura-Hamada, *Hot-carrier effects in MOS devices* (Academic Press, 1995)
52. Predictive Technology Model. http://ptm.asu.edu/
53. W.-K. Yeh, W.-H. Wang, Y.-K. Fang, F.-L. Yang, Temperature dependence of hot-carrier-induced degradation in 0.1 μm SOI nMOSFETs with thin oxide. IEEE Electron Device Lett. **23**(7), 425–427 (2002)
54. F. Cacho, P. Mora, W. Arfaoui, X. Federspiel, V. Huard, Hci/bti coupled model: the path for accurate and predictive reliability simulations. In *IEEE Reliability Physics Symposium, 2014 IEEE International*, 2014, pp. 5D–4
55. X. Chen, Y. Wang, Y. Cao, H. Yang, Statistical analysis of random telegraph noise in digital circuits. In IEEE Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific, 2014, pp. 161–166

56. T. Matsumoto, K. Kobayashi, H. Onodera, Impact of random telegraph noise on CMOS logic circuit reliability. In 2014 IEEE Proceedings of the IEEE Custom Integrated Circuits Conference (CICC), 2014, pp. 1–8
57. X. Wang, P.R. Rao, A. Mierop, A.J.P. Theuwissen, Random telegraph signal in CMOS image sensor pixels. In *IEEE Electron Devices Meeting, 2006. IEDM'06. International*, 2006, pp. 1–4
58. M. Yamaoka, H. Miki, A. Bansal, S. Wu, D.J. Frank, E. Leobandung, K. Torii, Evaluation methodology for random telegraph noise effects in SRAM arrays. In *2011 International Electron Devices Meeting*, 2011
59. K. Fukuda, Y. Shimizu, K. Amemiya, M. Kamoshida, C. Hu, Random telegraph noise in flash memories-model and technology scaling. In *IEEE Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, 2007, pp. 169–172
60. N. Tega, H. Miki, Z. Ren, P.D. Christoper, Y. Zhu, D.J. Frank, M.A. Guillorn, D.-G. Park, W. Haensch, K. Torii, Impact of HK/MG stacks and future device scaling on rtn. In *IEEE Reliability Physics Symposium (IRPS), 2011 IEEE International*, 2011, pp. 6A–5
61. N. Tega, H. Miki, F. Pagette, D.J. Frank, A. Ray, M.J. Rooks, W. Haensch et al., Increasing threshold voltage variation due to random telegraph noise in FETs as gate lengths scale to 20 nm. In *VLSI Technology, 2009 Symposium on IEEE*, 2009, pp. 50–51
62. K. Ito, T. Matsumoto, S. Nishizawa, H. Sunagawa, K. Kobayashi, H. Onodera, The impact of RTN on performance fluctuation in CMOS logic circuits. Population **50**, 100 (2011)
63. T. Matsumoto, K. Kobayashi, H. Onodera, Impact of random telegraph noise on CMOS logic delay uncertainty under low voltage operation. In *IEEE Electron Devices Meeting (IEDM), 2012 IEEE International*, 2012, pp. 25–6
64. M. Luo, R. Wang, S. Guo, J. Wang, J. Zou, R. Huang, Impacts of random telegraph noise (RTN) on digital circuits. IEEE Trans. Electron Devices **62**(6), 1725–1732 (2015)
65. L. Gerrer, J. Ding, S.M. Amoroso, F. Adamu-Lema, R. Hussin, D. Reid, C. Millar, A. Asenov, Modelling RTN and BTI in nanoscale MOSFETs from device to circuit: a review. Microelectron. Reliab. **54**(4), 682–697, 2014
66. K. Ito, T. Matsumoto, S. Nishizawa, H. Sunagawa, K. Kobayashi, H. Onodera, Modeling of random telegraph noise under circuit operation—simulation and measurement of RTN-induced delay fluctuation. In *2011 12th International Symposium on IEEE Quality Electronic Design (ISQED),* 2011, pp. 1–6
67. J.P. Campbell, J. Qin, K.P. Cheung, L.C. Yu, J.S. Suehle, A. Oates, K. Sheng, Random telegraph noise in highly scaled nMOSFETs. In *IEEE Reliability Physics Symposium, 2009 IEEE International*, 2009, pp. 382–388
68. S. Realov, K.L. Shepard, Analysis of random telegraph noise in 45-nm CMOS using on-chip characterization system. IEEE Trans. Electron Devices **60**(5), 1716–1722 (2013)
69. A. Ghetti, C. Monzio Compagnoni, F. Biancardi, A.L. Lacaita, S. Beltrami, L. Chiavarone, A.S. Spinelli, A. Visconti, Scaling trends for random telegraph noise in deca-nanometer flash memories. In *IEEE Electron Devices Meeting, 2008. IEDM 2008. IEEE International*, pp. 1–4 (2008)
70. K. Takeuchi, T. Nagumo, S. Yoko Gawa, K. Imai, Y. Hayashi, Single-charge-based modeling of transistor characteristics fluctuations based on statistical measurement of RTN amplitude. In *VLSI Technology, 2009 Symposium on IEEE*, 2009, pp. 54–55
71. N. Tega, H. Miki, Z. Ren, C. Emic, Y. Zhu, D.J. Frank, J. Cai, M.A. Guillorn, D.-G. Park, W. Haensch et al., Reduction of random telegraph noise in high-κ /metal-gate stacks for 22 nm generation FETs. In IEEE Electron Devices Meeting (IEDM), 2009 IEEE International, 2009, pp. 1–4
72. L. Gerrer, S. Maria Amoroso, P. Asenov, J. Ding, B. Cheng, F. Adamu-Lema, S. Markov, A. Asenov, D. Reid, C. Millar, Interplay between statistical reliability and variability: a comprehensive transistor-to-circuit simulation technology. In *Proceedings of Reliability Physics Symposium (IRPS) A*, volume 3, 2013

73. J. Martin-Martinez, J. Diaz, R. Rodriguez, M. Nafria, X. Aymerich, E. Roca, F.V. Fernandez, A. Rubio, Characterization of random telegraph noise and its impact on reliability of SRAM sense amplifiers. In *CMOS Variability (VARI), 2014 5th European Workshop on IEEE*, 2014, pp. 1–6
74. B. Zimmer, O. Thomas, S.O. Toh, T. Vincent, K. Asanovic, B. Nikolic, Joint impact of random variations and RTN on dynamic writeability in 28 nm bulk and FDSOI SRAM. In Solid State Device Research Conference (ESSDERC), 2014 44th European IEEE, IEEE, pp. 98–101
75. M. Choudhury, V. Chandra, K. Mohanram, R. Aitken, Analytical model for tddb-based performance degradation in combinational logic. In Design, Automation & Test in Europe Conference & Exhibition, 2010, *IEEE*, pp. 423–428, 2010
76. B. Kaczer, R. Degraeve, R. O'Connor, P. Roussel, G. Groeseneken, Implications of progressive wear-out for lifetime extrapolation of ultra-thin (EOT 1 nm) sion films. In *International Electron Devices Meeting*, pp. 713–716, 2004
77. K.F. Schuegraf, C. Hu, Hole injection SiO_2 breakdown model for very low voltage lifetime extrapolation. IEEE Trans. Electron Devices **41**(5), 761–767 (1994)
78. J.W. McPherson, H.C. Mogul, Underlying physics of the thermochemical e model in describing low-field time-dependent dielectric breakdown in SiO_2 thin films. J. Appl. Phys. **84**, 1513–1523 (1998)
79. D.J. DiMaria, E. Cartier, Mechanism for stress-induced leakage currents in thin silicon dioxide films. J. Appl. Phys. **78**(6), 3883–3894 (1995)
80. E.Y. Wu, J. Suné, Power-law voltage acceleration: a key element for ultra-thin gate oxide reliability. Microelectron. Reliab. **45**(12), 1809–1834 (2005)
81. E.Y. Wu, W.W. Abadeer, L.-K. Han, S.-H. Lo, G. Hueckel, Challenges for accurate reliability projections in the ultra-thin oxide regime. In IEEE/IRPS, pp. 57–65, 1999
82. K. Arabi, R. Saleh, X. Meng, Power supply noise in SoCs: metrics, management, and measurement. IEEE Des. Test Comput. **24**(3), 236–244 (2007)
83. S.R. Nassif, Power grid analysis benchmarks. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC)*, 2008, pp. 376–381
84. K. Haghdad, M. Anis, Power yield analysis under process and temperature variations. IEEE Trans. Very Large Scale Integr. Syst. (TVLSI) **99**, 1–10 (2011)
85. S. Mukherjee, *Architecture Design for Soft Errors* (Morgan Kaufmann, 2011)
86. P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, L. Alvisi. Modeling the effect of technology trends on the soft error rate of combinational logic. In *Proceedings of the International Conference on Dependable Systems and Networks*, 2002, pp. 23–26
87. B.D. Sierawski, J.A. Pellish, R.A. Reed, R.D. Schrimpf, K.M. Warren, R.A. Weller, M.H. Mendenhall, J.D. Black, A.D. Tipton, M.A. Xapsos et al., Impact of low-energy proton induced upsets on test methods and rate predictions. T-NS **56**(6), 3085–3092 (2009)
88. D.F. Heidel, P.W. Marshall, K.A. LaBel, J.R. Schwank, K.P. Rodbell, M.C. Hakey, M.D. Berg, P.E. Dodd, M.R. Friendlich, A.D. Phan et al., Low energy proton single-event-upset test results on 65 nm SOI SRAM. T-NS **55**(6), 3394–3400 (2008)
89. K.P. Rodbell, D.F. Heidel, H.H.K. Tang, M.S. Gordon, P. Oldiges, C.E. Murray, Low-energy proton-induced single-event-upsets in 65 nm node, silicon-on-insulator, latches and memory cells. *T-NS* **54**(6), 2474–2479 (2007)
90. http://en.wikipedia.org/wiki/Muon. Accessed 21 Jan 2015
91. J-L. Autran, S. Semikh, D. Munteanu, S. Serre, G. Gasiot, P. Roche, *Soft-error Rate of Advanced SRAM Memories: Modeling and Monte Carlo Simulation* (Numerical Simulation: From Theory to Industry, 2012)
92. F. Lei, S. Clucas, C. Dyer, P. Truscott, An atmospheric radiation model based on response matrices generated by detailed Monte Carlo simulations of cosmic ray interactions. IEEE Trans. Nucl. Sci. **51**(6), 3442–3451 (2004)
93. G.A. Sai-Halasz, M.R. Wordeman, R.H. Dennard, Alpha-particle-induced soft error rate in vlsi circuits. *T-ED* **29**(4), 725–731 (1982)
94. R.C. Baumann, Radiation-induced soft errors in advanced semiconductor technologies. IEEE Trans. Device Mater. Reliab. **5**(3), 305–316 (2005)

Dependability Threats

Cristiana Bolchini, Maria K. Michael, Antonio Miele and Stelios Neophytou

Abstract This chapter discusses dependability threads for modern integrated circuits that affect both their correct operation and performance. The text provides an overview of fault/error models adopted in methodologies for dependability assessment, analysis, and mitigation. Faults are categorized based on their applicability in the various abstraction layers. Their applicability to modern design trends such as FPGAs and NoCs is also presented. Furthermore, models for emerging and future dependability issues are discussed in the same rationale. In particular, special attention is given to those issues that typically arise during the operational life of the devices, causing either transient, intermittent or permanent failures, including aging and wear-out effects that directly affect their lifetime.

1 Introduction

The continuous technology advances have been triggering the evolution of integrated circuits, allowing remarkable potential usages of electronics. As the technology moves towards feature sizes of a few nanometers, the competence of the circuit to deliver the intended functionality is questioned. Different dependability threads resulting from the stringent design constraints, modern design approaches, as well as the increased susceptibility of such dense designs to environmental

C. Bolchini (✉) · A. Miele
Politecnico di Milano, DEIB, P.zza L. da Vinci, 32, 20133 Milano (MI), Italy
e-mail: cristiana.bolchini@polimi.it

A. Miele
e-mail: antonio.miele@polimi.it

M.K. Michael
University of Cyprus, 75 Kallipoleos Avenue, 1678 Nicosia, Cyprus
e-mail: mmichael@ucy.ac.cy

S. Neophytou
University of Nicosia, 46 Makedonitissas Avenue, 2414 Nicosia, Cyprus
e-mail: neophytou.s@unic.ac.cy

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_2

conditions affect both the robustness and efficiency of the circuits. These threads do not only affect the manufacturability of the circuit but constitute an ongoing vulnerability factor throughout its lifetime.

This new situation requires attentive approaches which, in order to ensure reliability, have to accurately assess and effectively mitigate the increasingly defecting behavior of integrated circuits. Although traditionally these important goals where achieved solely by considering processes like design verification, design for testability, testing, and diagnosis, in modern and future highly dense circuits this can be proven to be insufficient. In fact, there are credible indications that we are moving towards a new era where circuits have to become capable of operating correctly and with the expected performance in the presence of faults, whether these are the result of design errors, process variations or environmentally induced defects. Thus, the effective assessment and mitigation requirements are imposing the development of a fault-tolerant ecosystem that will be capable of detecting incorrect behavior and either use redundancy to nullify its effect or enable reconfiguration mechanisms to bypass the defective components.

This chapter begins by presenting the fundamental concept of fault modeling and its crucial role to the whole range of problems regarded in the assessment and mitigation of dependability threads (Sect. 2). Specifically, Sect. 2.1 discusses the different categories of faults defined by their nature, frequency of occurrence and lifetime. In Sect. 2.2 the faults that cause permanent damage to the circuit's function are discussed, while in Sect. 2.3 the main types of non-permanent faults are discussed together with their sources and corresponding mitigation techniques. Fault models and their usage in contemporary design approaches such as field programmable gate arrays and networks-on-chip are presented in Sect. 2.4.

Section 3 discusses aging and wear-out issues including modeling and reliability estimation. Sections 3.1 and 3.2 present the basic concepts for reliability characterization explaining the corresponding failure mechanisms and performing analysis of their distributions. In Sect. 3.3 this reliability model is extended to cover variable working conditions, while a comprehensive system-level model considering multi-component architectures is defined in Sect. 3.4.

Finally, in Sect. 4 relevant metrics are overviewed as the principal assessment tool to support decision-making in the design of methodologies and techniques targeting dependability problems.

2 Fault Models

The dependability of an electronic system can be impaired in several ways, typically referred to as *failures*, *errors*, *defects*, and *faults*. The common concept in all these four definitions is the fact that they describe some type of unexpected and unintended problem; however, each refers to a distinct level of the electronic system [1].

A *failure* is generally defined in terms of system-level operation and occurs when the system does not perform its specified or expected service. Hence, a system failure exists when the system *fails* to do what it should do. Failures can be caused by errors.

An *error* occurs at the computational level. Hence, an error is a deviation from correctness in computation which is ultimately captured by a wrong system output or system state value. An erroneous system can potentially lead to a failure. Errors can be caused by physical level defects as well as by various external factors, which are modeled as faults.

A *defect* occurs at the physical level and denotes some type of imperfection or flaw. It is the unintended deviation between the actual hardware and its intended design. Defects can appear during fabrication caused by a variety of sources such as process defects, material defects and package defects. Furthermore, defects can appear during the lifetime of the electronic system caused mainly by device aging and wear-out phenomena. A defective system can trigger the appearance of an error.

A *fault* is a representation (at some design abstraction layer) of a physical difference between the expected "correct" system and the system under consideration. The physical difference can be caused by physical defects but also by external factors, such as nuclear and electromagnetic radiation, temperature, vibration, etc.

It is important to understand that not all defects (nor external factors) can be represented as faults and, hence, some of these do not manifest themselves as faults. Furthermore, once a fault appears it does not necessarily lead to an error and, similarly, not every error causes a system failure.

This section elaborates on fault characterization and fault models which have been widely used to model the effects of physical silicon defects as well as certain external factors. The classical taxonomy of permanent, transient, and intermittent faults is considered and various fault models are discussed, taking into consideration the different layers of design model abstraction (e.g., transistor, switch, logic, RTL, microarchitectural and system level). Fault modeling in modern technologies such as Field Programmable Gate Arrays (FPGAs) and Networks-on-Chip (NoC) is also presented. The section concludes with a discussion on recent approaches considering fault models for cross-layer abstraction modeling, for the purpose of cross-layer reliability evaluation and fault mitigation.

2.1 Fault Classification and Modeling

2.1.1 Fault Classification

A fault can be characterized based on various criteria, such as its nature, extent, and duration. In terms of nature, faults are classified as logical or non-logical [2]. A logical fault represents the effect of a physical fault on the behavior of a modeled system and exists when it causes a logic discrepancy at some point in the system.

Non-logical faults include the remaining faults, such as power failures, clock signal malfunction, etc. The great majority of defects cause logical faults and only few are known to cause non-logical ones. Faults can also be classified as local or distributed based on the extent of their effect. The most popular classification of faults relates to their duration, which classifies faults as *permanent* and *non-permanent* according to the way faults manifest themselves in time.

Permanent faults are those whose presence affects the functional behavior of the system on a permanent basis. Permanent faults remain active until a mitigation action is taken, which typically deactivates the fault through bypassing or masking. The main source of permanent faults comes from physical defects in the hardware causing irreversible physical changes, such as shorts and opens in a circuit, broken interconnections, etc. Another source, not as frequent as that of physical defects, is that of functional design errors which are caused by incorrect functional design implementation. Up until recently, permanent faults created by these two sources were expected to be identified during the validation/verification and manufacturing test processes and, hence, the appearance of a permanent fault during the lifetime of a system was considered a rear event. Unfortunately, due to the increased complexity of ultra-large integration and the continued device scaling, current design implementations are becoming less verifiable and fabrication technologies less reliable, respectively. In order to maintain acceptable yield, products with some permanent faults can be used in field, in contrast to practices of the past where such products were discarded. Furthermore, the combination of increased process variability with new phenomena accelerates the aging of electronic devices (discussed in detail in Sect. 3) and escalates the problem as they can lead to the creation of permanent faults during the system's lifetime. As a result, permanent faults are now expected to exist and continue to increase during the lifetime of an electronic system and their detection and mitigation have become extremely important in ensuring the reliable operation of the system.

Non-permanent faults are temporary in nature in the sense that they are not present at all times. These faults occur mainly in a random fashion and affect the functional behavior of the system for finite, but often unknown, periods of time. Non-permanent faults are further classified into *transient* and *intermittent* faults. Nanocomputing devices are expected to experience increased error rates due to transient and intermittent faults [3]. The main distinction between the two classes of faults is their origin.

Transient faults are typically caused by environmental conditions, most often by α-particles and cosmic rays, but also by temperature, humidity, pollution, pressure, vibration, electromagnetic interference, power supply fluctuations, static electric discharges, among others [4]. Their appearance is random, non-recurring, and not expected to cause any type of permanent damage. The most frequent effect of a transient fault is referred to as soft error. Different error models for soft errors are discussed in detail in Sect. 2.3.

Intermittent faults are not environmentally related. Instead, they occur due to unstable or marginal hardware caused by manufacturing residues which commonly include process variation or in-progress aging/wear-out, combined with temperature and voltage fluctuations, among others [5, 6]. Results from recent studies suggest that the dependability threats imposed by intermittent faults are significant and increasing in nanocomputing technologies [5]. A main characteristic of intermittent faults is that, for the duration that they appear, they tend to behave similarly to permanent faults. This implies that similar fault models can be considered for both cases, perhaps under different scenarios. In contrast to transient faults which cannot be fixed by repair and require constant monitoring and mitigation, intermittent faults can be eliminated by replacing the impaired hardware, similarly to what happens when permanent faults exist. To distinguish intermittent faults from permanent and transient ones, three criteria can be used:

(i) Appearance of fault: Intermittent faults occur in burst with highly varied duration each time they appear.
(ii) Location of fault: An intermittent fault occurs at the same location, when the fault is activated.
(iii) Repeatability of fault: An intermittent fault is considered non-repeatable, it appears only under particular situations.

2.1.2 Fault Modeling

The effect of a fault is represented by a model, which represents the change produced by the fault in the system's behavior, referred to as fault model. In general, fault modeling bridges the gap between the physical reality of defects and the underlying mathematical abstraction model used to represent the system. A fault model identifies the targets to be considered, allowing the application of analytical tools to effectively analyze and measure the effects of faults. Fault modeling reduces considerably the complexity of defect analysis, as many defects can be modeled by the same logical faults. Also, several logical fault models are technology-independent, which helps in eliminating the necessity for distinct technology-dependent fault models.

In practice, modeling of faults relates closely to modeling a system. In the design hierarchy, a system can be represented by different abstraction models based on the layer of the hierarchy considered. The higher a model is in the design hierarchy, the fewest implementation details it contains (e.g., behavioral and functional models), hence, it imposes smaller complexity for representation and simulation purposes. On the other hand, models towards the bottom of the design hierarchy (e.g., switch and geometric models) include more accurate implementation details and correlate stronger to the actual hardware in the expense of additional model complexity. Fault modeling works in a similar philosophy. Fault models at the lower layers correlate better to actual physical defects and, thus, are more accurate. However, due to the high complexity of today's systems, it is often necessary for practical reasons to consider fault models at higher layers which are not as accurate.

Certain fault models do not fit in any of the design hierarchies. These consider usually *defect-oriented faults*, which represent physical defects not appropriately captured by any other fault model and for this reason they are often referred to as "realistic" fault models. Quiescent current faults (I_{DDQ}) which model leakage of quiescent current in CMOS gates is such an example as well as some analog fault models.

The next subsection discusses major fault models for permanent faults based on the related abstraction layer of the model. As discussed above, intermittent faults can be modeled as permanent ones; however, they required different scenarios for detection. Section 2.3 deals mainly with transient faults caused by soft errors and presents the various typically adopted error models.

2.2 *Fault Models for Permanent Faults*

Fault models for permanent faults have been developed for each layer of abstraction such as behavioral, functional, structural, switch, and geometric. This subsection presents the main fault models used at each layer. For a more extensive list and detailed description of fault models the interested reader is referred to [4, 7, 8].

2.2.1 Behavioral Fault Models

Behavioral fault models consider the behavioral specification of the system, as described by a high-level programming language, such as C/C++/SystemC, or a hardware-description language, such as VHDL/Verilog. As a behavioral layer description is often at the top of the design hierarchy of a system, the terms behavioral and system are typically used interchangeably. However, it is important to note that an electronic component/system can have a behavioral representation at different abstraction layers, such as in the form of Boolean equations describing gates (at the structural layer) or data flow descriptions for registers (at the RTL layer). Regardless of the underlying abstraction layer, behavioral fault models are derived based on the procedural descriptions where a fault relates to some type of incorrect execution of the constructs of the descriptive language used. The precise nature of the faults included in a behavioral fault model is determined by the acceptable complexity of the model, and how well these faults correlate to more accurate fault models described at lower layers of the abstraction hierarchy (e.g., structural fault models). An extensive discussion on behavioral fault models can be found in [9–12], among many others.

The various behavioral fault models consider possible failure modes on the constituent language constructs. Some of the most well-known behavioral fault models used for design validation, fault simulation, and test vector generation are derived based on the following failure modes:

- Permanent *variable* values, representing the lower and upper extremes of the corresponding data type of a variable. Such failures are captured by the **behavioral stuck-at fault model**;
- Failures in *function calls* where the same value is always returned by a function, corresponding to the lower and upper extremes of the function range;
- Failures in various statements resulting in erroneous program *control flow,* such as loops and branching statements. Examples include *for*, *wait for*, *switch*, *case*, and *if-then-else* statements. These are often referred to as **branch faults**;
- Failures in variable/signal assignment statements captured by the **behavioral stuck-open fault model**. In this case, the value of the target does not follow that of the source expression in an assignment statement;
- *Micro-operation* failures refer to deviations of an operator, arithmetic or relational, from its intended functionality. In this case, the operator may be faulted to any other operator in its class. The **micro-operation fault model** considers perturbation among certain classes of operators;
- *Assertion* failures occur when assertion statements do not hold. Hence, an **assertion fault** implies that the corresponding property is not "true" indicating a possible problem.

The effectiveness of behavioral fault models can be evaluated based on software test metrics, such as *statement coverage* and *branch coverage*. Moreover, to validate the models it is necessary to investigate the correlation to actual defects in the corresponding hardware realizations which is usually done by fault simulations at lower layers of abstraction.

2.2.2 Functional Fault Models

The purpose of a fault model at the functional layer is to ensure that the functional behavior of a given block/component or circuit/system is consistent with the expected function of the design. Functional fault models are typically based on the input/output relationship of higher level primitives which may incorporate a large number of gates. Often, these fault models are ad hoc as they are derived from specific functional blocks or microprocessor designs at RTL, as described below. Similarly to behavioral fault models, the effectiveness of functional fault models depends on their ability to provide high fault coverage when lower-level fault models are considered. This is more of a challenge in models using large functional blocks or microprocessor models. For the case of memory blocks, various well-established functional fault models exists which have been demonstrated to have strong correlation to actual permanent defects.

Functional Block Fault Models

This category includes specific models for small functional modules, which are usually derived in an ad hoc manner. Due to their low complexity, these models are easy to handle and typically provide high coverage. Popular modules with such corresponding functional fault models include multiplexers, single-bit full-adders, decoders, etc [4, 10]. As an example, consider the following functional fault model for an n-to-2^n decoder, which was shown to cover all physical shorts and opens in the switch-layer design (transistor-level implementation) [13]:

(i) Only an incorrect output line is activated;
(ii) Both a correct and an incorrect lines are activated;
(iii) No output line is activated.

Such models for different small functional modules are often the basis for creating functional fault models for larger functional blocks, including iterative logic array (ILA) designs. ILAs are block-based designs, where a block (logic cell) is replicated multiple times and interconnected appropriately. A very popular example of an ILA design is the n-bit ripple-carry adder which consists of exactly n 1-bit full adders. In this case, the **single cell fault model** considers each of the 1-bit full adders as a cell and assumes that at any time only a single cell can be faulty. The possible faults considered in a single cell can be those derived from the functional block fault model of the constituent 1-bit full adder. Additional faults can be considered in ILA-based fault models to account for possible faults in the interconnection logic between the cells.

This category also includes generalized fault models for functional blocks based on the truth table of the block. The **general functional fault model** (GF) is mainly practical for small functions as for an n input function it exhaustively considers all 2^n possible faults [14]. The extension of the GF model to sequential circuits was shown to represent specific types of pattern-sensitive faults in RAMS and faults in simple bit-sliced microprocessors [15]. Several methods have been proposed to reduce the complexity of the GF model and make it applicable to larger blocks, but often at the expense of reduced coverage or restrictions on the implementation of the function. One such example is the *universal test set*, used primarily for manufacturing test purposes, which reduces significantly the number of tests needed to check for any fault changing the truth table provided that the function is synthesized using only the universal set of gates (AND, OR, NOT) [4].

Another interesting family of fault models is that of **physically induced** functional fault models [16]. In this case, more “physical” faults (i.e., faults at lower abstraction layers such as structural and switch) are strongly correlated to functional faults. More precisely, given a full set of physical faults, a set of functional faults is derived. Complete coverage of the functional faults also provides complete coverage of the correlated lower-level faults. Different physical fault models can be considered, including gate-level single stuck-at faults, bridging faults, and switch-level faults. Similarly to physically induced fault models, certain gate-level delay faults can be modeled using the **single-input change** and the **multiple-input change** functional fault models [17]. These models are less complex; however they do not provide completeness in fault coverage between the two considered models.

Microprocessor Fault Models

As microprocessors are among the most complex circuits, their modeling at lower abstraction layers (such as structural and below) can be prohibitive for fault representation and various related tasks such as fault simulation. Hence, functional modeling is typically utilized for microprocessors, most often at the register transfer level (RTL). The seminal work in [18] proposed various fault models based on a set of functions representing a microprocessor. These functions include register decoding, data transfer, data manipulation, and instructions sequencing. For each of the functions a corresponding fault models exists. Later work in [19] proposed a refined fault model based on the specific microprocessor instruction set, known as the **instruction fault model**. In this case, each instruction comprises of multiple micro-instructions each of which is composed of a set of micro-orders. An instruction fault in this combined microprocessor fault model is assumed to exist in any of the following cases:

(i) When one or more micro-orders of an instruction do not execute;
(ii) When additional, non intended, micro-orders are executed with an instruction;
(iii) When both (i) and (ii) occur for the same instruction.

In order to study the effect of hardware faults at the system level, the recent work of [20] considers the **microarchitecture-level stuck-at** fault model, which models logic stuck-at faults at the inputs and outputs of latch elements, and proposes two new probabilistic functional models, the **P-model** and the **PD-model**. All three models were used to estimate the effect of gate-level faults (both stuck-at and delay) and were found to be inaccurate in capturing the system-level effects of gate-level faults, demonstrating the difficulties and complexities in deriving meaningful fault models for permanent hardware defects at the microarchitecture level. A possible solution to this problem is mixed-mode fault models, such as the one considered in [21], where both gate-level and RTL descriptions are used to model a microprocessor leading to better accuracy in capturing challenging lower-level fault models such as the gate-level path delay fault model at RLT.

Memory Fault Models

Functional fault models are widely used for memory components, allowing for generalized methodologies independent to the specific underlying technology used to implement the memory. As random-access memory (RAM) has been the predominant chip design for memory, the standard RAM functional model has been considered [22], for both dynamic RAM (DRAM) and static RAM (SRAM) designs. Moreover, this functional block-based model is easily transformed to other categories of memory such as read-only memory (ROM) and various types of programmable memory. A plethora of functional fault models exists based on this model, considering faults in the constituent blocks as well as their interconnections. Very often, the model can be further simplified by reducing the various blocks into three main blocks: (i) the address decoder block, including the row and column decoder units and the address latch, (ii) the read/write logic block, including the write driver, sense amplifier and necessary data register, and (iii) the memory cell block consisting of the memory cell array. The address decoder and read/write blocks have specific fault models, as described above in functional block fault models. Below we concentrate on popular fault models for the memory cell block of the RAM model, which are also used for other on-chip memory blocks such as caches and microprocessor memory arrays.

The early work of [23] proposed fault models for single stuck-at faults, transition faults and coupling faults. According to the reduced functional fault modeling analysis of [24], these three types of faults, in addition to a special type of coupling faults called pattern sensitive faults [25], are sufficient for memory cell blocks. A brief discussion on each of these fault models is given below:

- Under the **single stuck-at fault model** (SSA), a memory cell may be permanently stuck at the 0 or 1 logic value and, hence, the cell is always in the faulty state and cannot be changed. The notation $\langle\forall/0\rangle$ and $\langle\forall/1\rangle$ is used to denote that for all possible actions on the memory location the response is always 0 and 1, respectively;
- A **transition fault** (TF) at some memory cell exists when the memory cell cannot make the transition from 1 to 0, leading to a *down-transition fault* denoted as $\langle\downarrow/1\rangle$, or the transition from 0 to 1, leading to an *up-transition fault* denoted as $\langle\uparrow/0\rangle$. Some transition faults can be reduced to stuck-at faults;
- A **coupling fault** (CF) exists between two memory cells *i* and *j*, when a transition in cell *i* (from 0 to 1 or vice-versa) results in an unintended transition in cell *j*. This type of fault is known as a **2-coupling fault** as it involves 2 cells. The **k-coupling fault model** is an extension, and the **inversion coupling fault model** and the **idempotent coupling fault model** are special cases of the 2-coupling fault model.
- The **pattern sensitive fault model** (PSF) is the most generalized version of the k-coupling fault model, as in a PSF the contents of a memory cell *i* (or the ability of the cell to change values) is affected by the contents or transitions of *all* other cells. To restrict the complexity of this model the **neighborhood PSF model** (NPSF) only considers a pre-defined neighborhood of cells close to the base cell *i* which can affect *i*.

The above models remain relevant to recent and future memory technologies, as long as a memory cell array is modeled as a functional block arranged in the typical row-column format (including single-dimension arrays).

2.2.3 Structural Fault Models

Fault models at the structural abstraction layer are better correlated to actual defects, caused by the manufacturing process or by aging of electronic components during the normal life of a chip, as the gate-level description and the interconnections among the gates are known. The challenge of these models is their increased complexity making their applicability difficult in complex circuits, such as microprocessors, and large-scale systems such as current SoC systems, for fault simulation and reliability evaluation purposes. In such cases, mixed-layer models can be used where some parts of the system have a functional description and others are represented by structural models.

The most popular structural fault models used to represent faults relating to the functionality as well as the timing of structural descriptions are briefly presented below. The reader is referred to [4, 7] for in-depth analysis of the various fault models.

Stuck-At Fault Model

The assumption here is that a representation exists of interconnected Boolean gates and other primitive components (such as latches and flip-flops). In the **single stuck-at fault model** (SSF), any interconnection can exhibit two types of logical faults, a *stuck-at-1* (SA1) and a *stuck-at-0* (SA0) fault. Under this model, only a single fault can exist at one time. Stuck-at faults have been shown to have strong correlations to multiple realistic defects, such as opens and shorts in different device locations, and for this reason they are widely used to represent real defects at the logical level. The SSF model has complexity linear, in terms of the number of faults considered, to the size of the Boolean netlist which can be reduced by at least 50% when the concepts of *fault equivalence* and *fault dominance* are applied [7].

Fault Models for Sequential Components

Components with memory elements, such as flip-flops and latches, must be examined for some additional types of faults, on top of SSFs. The first group of faults, know as **initialization faults**, can interfere with the initialization process of flip-flops rendering the circuit incapable to initialize at a certain state or any state. Initialization faults can be triggered by SA1 or SA0 faults, however, their effect is different and they required special treatment. SSFs can also cause the occurrence of **star faults**, which include **oscillation faults** created by oscillating signals due to combinational feedback in the sequential logic, and **race faults** appearing in asynchronous sequential logic models and causing the well know problem of race conditions.

Delay Fault Models

Ensuring the temporal correctness of today's circuits and systems has become a major challenge due to the demand for increasingly complex designs and tighter timing constrains, in order to achieve the highest possible speed. This challenge exists both during the manufacturing process, in which case it is tackled by delay fault testing and diagnosis, and, equally important, during the normal life time of current and future technologies due to the appearance of aging and wear-out related physical mechanisms which can increase/modify the delay of the electronic devices (see Sect. 3).

Delay faults increase the combinational delay beyond the determined clock period, causing timing violations to appear. The most important delay fault models at the structural layer are presented below [26, 27]. This is an appropriate layer for modeling various types of delays, while maintaining reasonable complexity in the fault models. This complexity depends of the types of faults as well as the underlying delay model considered. More complex models offer higher accuracy in modeling actual delay defects. If accuracy is of great importance, fault models at the geometric layer are better suited but come with the cost of significantly increased complexity.

- The **transition delay fault model** (TDF) considers two types of faults at the output of each gate: the *slow-to-rise fault* and the *slow-to-fall fault*. The TDF model is widely used by the industry for manufacturing delay testing, primarily due to its low complexity. The basic underlying assumption in this model is that a single gate delay becomes large enough to violate the system nominal delay value independently from which path the delay propagates. For this reason, the TDF is also referred to as the **gross gate-delay fault model**. When this assumption does not hold, the TDF model often fails to identify nominal delay violations caused by cumulative delays. Various models have been proposed to tackle this problem while maintaining the linear complexity of the TDF model, such as the one used in [28] which insists in propagating a transition fault via the most critical path(s) passing through the gate under consideration.
- The **path delay fault model** (PDF) is a more comprehensive model than the TDF model since it considers distributed propagation delays along one or multiple combinational paths in a circuit [29]. In this manner, both *lumped (gross) gate-delays* as well as *small gate-delays* can be captured adequately. Therefore, the PDF model can consider the cumulative effect of all delay variations of the gates and interconnections along one or more paths. A combinational path is one that originates at a primary input or the output of a clocked memory element and terminates at a primary output or the input of a clocked memory element, through a connected chain of logic gates. A single path delay fault appears if the propagation delay of a signal transition along a combination path exceeds the clock period. Hence, for each such path, two possible path delay faults are considered corresponding to the propagated rising and falling transitions at the origin of the path. Clearly, the complexity of the single PDF model is a major challenge as, in the worst case, it can be exponential to the number of interconnections in the netlist. Various delay fault models have been proposed in an attempt to alleviate this problem. The **segment delay fault model** is one such example as it considers a limited number of path segments, instead of all complete paths, making it more accurate than the TDF model while reducing the complexity of the PDF model. The **critical PDF model** which considers only complete critical paths, as determined by timing analysis, is perhaps the best compromise between accuracy and complexity.

2.2.4 Switch Fault Models

Fault models defined at the structural layer of abstraction, such as the SSF model, can still fail to capture some particular real defects. In these cases, lower abstraction representations must be considered. One such well-known situation involves certain transistor faults in MOS technologies, in particular in CMOS technology, which can be captured only if a switch layer model is considered. Two popular fault models are used for transistor faults, representing transistor opens and shorts:

- The **stuck-open fault model** considers the case where a transistor has become permanently non-conducting due to some defect between the source and the drain of the transistor, behaving as an open device. The worst effect of a stuck-open fault is the generation of a floating state at the output of the corresponding logic gate. These faults do not occur frequently, only about 1% of the CMOS faults are attributed to stuck-open faults, but once they appear their effect can be significant as they can transform a combinational component into a sequential one.
- A more frequent CMOS failure mechanism involves device shorts modeled as stuck-on faults (also referred to as stuck-short faults). The **stuck-on fault model** assumes that a transistor may be stuck permanently in the conducting phase due to a closed path between its source and its drain. When such a fault is activated, the state of the corresponding logic gate can vary, depending on the impedance of the transistor and the switching threshold of other devices in the logic gate.

2.2.5 Geometric Fault Models

At this layer, knowledge of the exact layout of the design under consideration is necessary in order to derive the possible defects that can occur. This includes device geometries, line widths, line separation distances, etc. In practice, two types of defects are considered, shorts and opens. The most well-known fault model at this layer is the **bridging fault model** (BF), as it addresses defects resulting from the continuous shrinking of geometries. Under this model, a *bridging fault* is a short between two signals. The fault universe in the exhaustive version of the BF model considers all possible pairs of signals. As this is not practical, as well as unrealistic, the possible pairs of signals to be considered can be derived by close examination of the layout. BF models also exist at higher abstractions layers such as the switch and structural layers. Clearly, the more information is available regarding the implementation the more accurate the model is. In this case, modeling BFs at the geometric layer can also help in reducing the complexity of the model (number of BFs considered), as the corresponding models at higher abstraction layers may consider a higher number of BFs in order to account for the low accuracy in modeling BFs.

BFs are categorized into *input* BFs or *non-input* BFs based on the location of their constituent signals. If two or more input signals of the same gate are shorted then this is an input BF, otherwise the fault is a non-input BF. Furthermore, BFs are classified as *feedback* BFs or *non-feedback* BFs. A feedback fault exists when the appearance of the fault creates a new feedback path in the circuit, otherwise the fault is non-feedback. Feedback faults are especially critical as they can cause oscillations.

Given a BF, the logic value of the shorted net is modeled in one of the following three ways:

(i) 1-dominant, created by an OR bridge (also known as wired-OR);
(ii) 0-dominant, created by an AND bridge (also known as wired-AND);
(iii) Indeterminate.

Case (iii) occurs mainly in CMOS technologies, where not all shorts can be mapped to a wired-AND or a wired-OR fault. If available, current monitoring is used for these cases; otherwise, both logic values are considered.

2.2.6 Multiple Fault Models

In contrast to the single fault models described above, multiple fault models consider the simultaneous presence of a set of single faults of the same type. Unless restricted, such fault models can exhibit an exponential number of faults making their applicability impractical. Due to their high complexity as well as the relatively small probability of occurrence, multiple fault models were not utilized frequently in the past. However, the continuous and aggressive technology shrinking has led to the increase of the probability of appearance of multiple faults, both at fabrication time and in-field time. Therefore, for both phases it is becoming increasingly necessary to consider multiple faults.

Multiple fault models are defined on different abstraction layers, similarly to the single fault models. Some of the most important categories of multiple fault models are presented below. For each case, the specific modeling abstraction layer is identified.

Multiple Stuck-at Fault Model

This refers to the most well known structural fault model, the single stuck-at fault model, when extended for multiple faults. A **multiple stuck-at fault** (MSF) is a condition under which a set of single stuck-at faults (of any cardinality) occur at the same time. Given this definition, it is clear that the number of multiple stuck-at faults is exponential to the number of interconnections in the gate-level netlist. In particular, there are $3^n - 1$ multiple faults in a model with n interconnections. Given this complexity and the fact that tests for single stuck-at faults have been shown to also detect multiple stuck-at faults, this model is not regularly used by the manufacturing test industry. However, the model is still necessary for diagnosing certain multiple faults and for testing designs with logic redundancy with high accuracy. Most importantly, the model becomes relevant when studying the reliability of current and new technologies which can experience multiple permanent faults during their normal lifetime.

Multiple Delay Fault Models

The structural layer path delay fault (PDF) model has been long considered as the most accurate one among traditional delay fault models (such as the transition fault model), due to its ability to detect both lumped as well as small distributed delay defects. The PDF set consists of single as well as multiple faults. **Multiple PDFs** (also referred to as functionally sensitized faults in the literature), model simultaneous excessive path delays among multiple paths and manifest only if all paths in the multiple fault exhibit delays. The number of multiple PDFs is intractable, even in the case of the **primitive PDF model** which is a refinement of the traditional PDF model and limits significantly the number of multiple faults needed to be considered [30, 31]. A more practical model is that of **critical primitive PDFs** which limits the number of overall faults considerably by considering only critical paths [32].

Multiple Memory Fault Models

Functional layer multiple memory fault models are nowadays employed as, due to the high density of the memory cell array, multiple faults are appearing in higher and increasing rates. *Linked* faults refer to multiple faults in which one fault influences the behavior of other faults, whereas, *unlinked* faults do not exhibit such behavior. Faults in both the **multiple SSA** and **multiple TF** models are unlinked. For the case of **multiple CFs**, both types of behavior can appear. The case of linked CFs is particularly challenging due to the effect of *fault masking*.

2.3 Transient and Intermittent Fault Models

Another important category of threads imposed during the lifetime of an integrated circuit produce errors which are by nature not permanent that may or may not be catastrophic for the device within they occur or for the entire system. This depends both on the severity of the thread as well as the application used. For example consider a computing system responsible for a space rocket launch. A non-permanent error hitting the system a few minutes after the successful takeoff can result in the rocket crash (e.g., because it had altered the procedure of leaving the earth atmosphere) even though the error was not severe and had a short life. Non-permanent errors are a subset of *Single-Event Effects* (SEEs) and are more usually referred to as *Soft Errors*. They are mainly caused by alpha particles and neutron (originated from cosmic rays) strikes on circuits.

2.3.1 Origins of the Soft Errors

While soft errors were introduced as a digital systems reliability thread from their early realization, before 1970s it was believed to be specific for space applications where the radiation energy is high. Intel Corp. was the first to report soft errors due to alpha particle strikes in dynamic RAM devices when their packaging accidentally used water contaminated by a radioactive element [33, 34]. Many other such cases have been reported ever since, enforcing semiconductor companies to incorporate safeguards to keep radiation as low as possible in their manufacturing process. Specifically, companies' actions toward this direction are twofold: (i) high purification of materials in order to minimize emission of particles from radioactive material traced in packaging such as uranium and thorium and (ii) techniques to minimize the probability an alpha particle reaches a device [35]. Modern packaging material specifications enforce alpha particle emission rates to be below $2 \times 10^{-3}\alpha$ counts per hour per cm^2 [35].

Nevertheless, the soft error thread is valid during the component's entire lifetime, even after delivered to the end user. This thread is not only due to the packaging impurities. Cosmic radiation interacts with the earth's atmosphere triggering a cascading effect that results in particle flux which is visible by and may affect the operation of integrated circuits in terrestrial systems [36–38]. Soft errors can arise from process variations as well, yet they are considered as a less important origin of faulty behavior.

Several studies have shown that cosmic radiation at ground level affects the operation of random-access memories (RAMs) [39, 40]. Hence, mitigation techniques such as error correction codes (ECC) and data interleaving have become a necessity especially in systems that require high reliability [41, 42]. In addition, shrinking technology has made combinational logic as a potential reliability victim.

Figure 1 illustrates the strike of a high-energy particle on a CMOS transistor in order to demonstrate its effect in the containing device functionality. Once a high-energy particle (in the range of 4–9 meV) strikes a semiconductor device, it disturbs its crystalline structure, especially the reverse biased junctions. The particle penetrates the device and creates a high carrier density track that consists of pairs of electrons and holes. This disturbs the depletion region expanding it deeper into the substrate. The diffusion creates current/voltage transients at the device which recedes as a function of time. Yet this transient condition results to a temporary change in the functionality of the device that may be observed as a change of a logic value at the netlist level of abstraction, i.e., a soft error.

As the transistor density is increased and the voltage levels are decreased, the energy of a particle becomes adequate to produce a fault that may lead to a failure of the corresponding system [43, 44]. The minimum charge of a particle that can cause a fault to a specific circuit device is known as *critical charge* (Q_{crit}) [34]. This charge must be computed for each device and technology process independently, usually using simulation tools. Obviously, the higher this charge the smaller the thread for the corresponding device. Q_{crit} is directly proportional to the operational

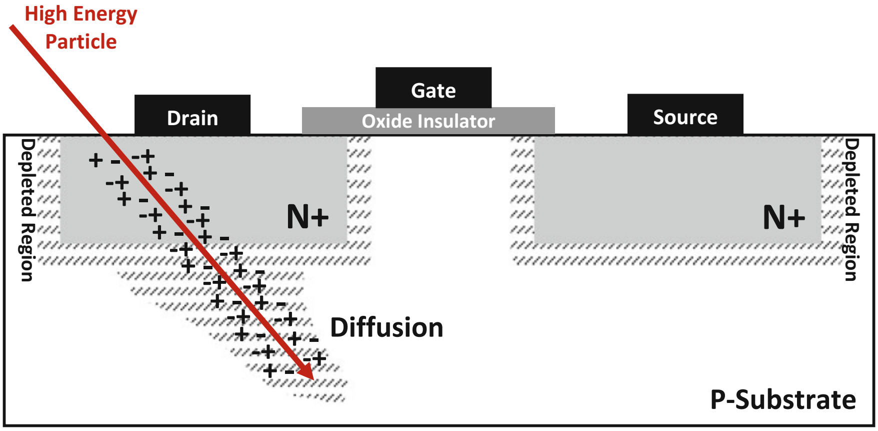


Fig. 1 A particle strike effect on a transistor

voltage of the device as well as its capacitance. Moreover, a number of electrical phenomena, such as power density and current density are increased, negatively affecting the operation of the circuit. In many cases such changes in the operational conditions of a component may transform a transient error into intermittent or, even worse, into a permanent one.

Consequently, soft errors are treated as radiation effects and are primarily characterized by the *Soft Error Rate* (SER), a metric that defines the mitigation approach to be followed for each of type of it. A typical metric for SER is *Failures in Time* (FIT) indicating a failure per billion operational hours. The FIT of a system can be calculated as the sum of the FIT values of the individual components of the system since the failures due to soft errors in different components are in principle unrelated.

2.3.2 Types of Soft Errors

The nature of transient SEEs defines the type of these errors and can help in predicting its expected behavior. Here, we provide a description for the most important soft errors and their main characteristics.

Single-Event Upset

A **Single-Event Upset** (SEU) arises when a particle strikes a memory element altering its state. Hence, they usually result in **Single-Bit Upsets** (SBUs) and they are treated under this assumption. Such strikes occur inside a RAM structure or on a flip-flop and their interaction with the combinational logic within certain time interval may further propagate the fault and cause an error. Also, an SEU can be an effect of an erroneous calculation within the combinational part of the circuit that is propagated and stored in a latch or flip-flop (described below as Single-Event

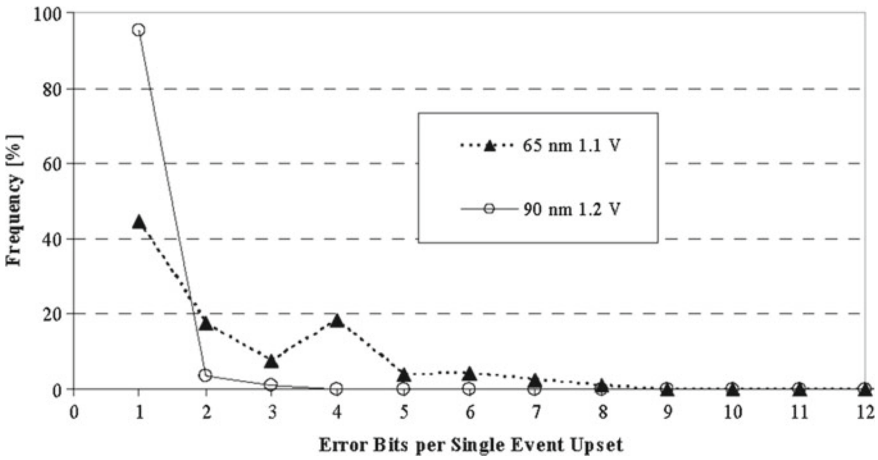
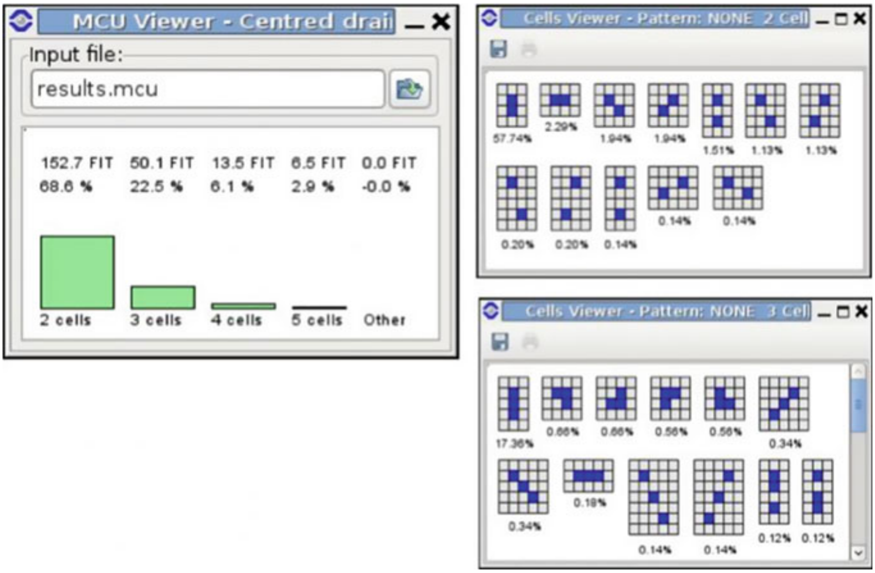


Fig. 2 Frequency distribution of number of faulty bits generated from a single-event upset for two different process sizes [45]

Transient). SEUs are usually restrained by single error correction codes (ECC) to cancel the effect of the upset as early as possible within the part of the circuit that occurred. ECC incorporate additional data bits to indicate the bit upset location and, consequently, correct it. Yet as the bit density is increased, the single-bit upset assumption becomes unrealistic [45, 46]. Figure 2 from [45] illustrates this trend for two different process technologies, i.e. 65 nm with supply voltage 1.1 V and 90 nm with supply voltage 1.2 V. In the older process generation, where the feature size and the supply voltage are larger, the vast majority of faults (>95%) affect only a single memory bit. In the newer generation this percentage drops a lot (~45%) rising the corresponding percentage for double, triple and quadruple error bits to considerable levels.

Multi-Cell Upset and Multi-Bit Upset

As a result of technology shrinking, Q_{crit} decreases giving rise to emerging types of upsets causing multiple disruptions where the SEU affects more than one memory position, either adjacent cells/flip-flops or different bits within the same memory word. In these cases, data correction requires more sophisticated and expensive mitigation methods, even beyond ECC. While the correlation between SEUs and SBUs is straightforward, a particle strike can affect more than one single bit in a memory element, in close proximity. The affected bits usually follow a pattern similar to those shown on the right part of Fig. 3. When the memory corruption occurs in adjacent memory cells or flip-flops then the fault is known as **Multi-Cell Upset** (MCU). If the corruption is on different bits of the same data word the fault is known as **Multi-Bit Upset** (MBU). Distinguishing between MCUs and MBUs is necessary in order to develop appropriate mitigation techniques for these two cases. For example, in order to effectively reverse the effect of MBUs, bits from different



Input file: results.mcu

Fig. 3 MCU vulnerability analysis as obtained from tool TFIT from IROC Tech [49]

data words are interleaved to keep physically apart bits from the same word. Appropriate software tools can be used to estimate how vulnerable memory cells are to SEUs. The output of such a tool is shown in Fig. 3. The tool, called TFIT^{TM} and developed by IROC Tech [47–49], evaluates the FIT of a given design under a specific technology process considering the MCU fault model. In addition to reporting FIT estimation for the number of cells affected per SEU (left part of Fig. 3) it can provide detailed FIT for different MCU patterns, as shown on the right part of Fig. 3.

Single-Event Transient

A voltage glitch in the circuit can be the reason of an undesired situation known as Single-Error Transient (SET). SET changes the operation of a logic gate that produces erroneous values. When these errors are propagated in a latch or flip-flop an SEU is triggered. It is expected that a large percentage of SETs are masked either logically or electrically before they can cause a change in the state of a circuit and, in turn, a system failure. In addition to that, in many cases the duration and/or the amplitude of the glitch is not sufficient to cause an SET, yet these cases will become minority as the technology scales further. Figure 4 illustrates the occurrence of an SET in a logic circuit. The error occurs at the combinational logic, altering the output F of the OR gate for a small time interval. This error is propagated to the input of a memory element (D Flip-Flop) arriving at the same time as the clock, and lasting long enough to change the state of the flip-flop which, in turn, affects the execution of subsequent time frames. Although the result of an SET is similar to

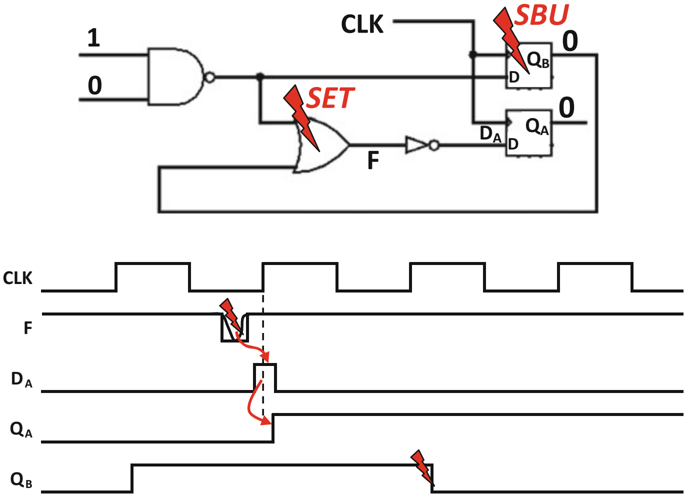


Fig. 4 Difference between single-event transient (SET) and single bit upset (SBU)

that of an SBU, their cause is not the same. Figure 4 also illustrates the occurrence of an SBU which, as opposed to SET, directly strikes the memory element Q_B. Thus, an SBU definitely results in an SEU, while SET may result in one or more SBUs (when there exist more than one propagation paths to memory elements) under certain conditions. Discrimination between SET and SBU is necessary for the development of appropriate mitigation techniques. Mitigation techniques consider redundant circuitry similar to those used for permanent faults. SET filtering techniques have been also proposed, utilizing duplication or signal delaying to cancel the specific characteristics of this event [50, 51].

Mitigation techniques consider redundant circuitry similar to those used for permanent faults. SET filtering techniques have been also proposed, utilizing duplication or signal delaying to cancel the specific characteristics of this event [50, 51].

Single-Event Latchup

An SEE occurring on a semiconductor structure may convert it to a current rectifier of bipolar nature. This fault is known as Single-Even Latchup (SEL) since it has similar characteristics to electrical latchup. The particle strike (usually by neutrons or protons) affects the parasitic structures of the PN junctions to trigger a situation that can be understood as two coupled bipolar transistors forming a parasitic thyristor (PNPN junction) [36, 38]. This situation may produce high currents which in turn increase the power dissipation, and thus, the temperature of respective component. When the temperature is increased beyond the thermal budget of the component (or the system) it can lead to a failure and, in some cases, create a permanent damage. Early detection mechanisms can identify such problematic situations and initiate a power reset of the system to prevent the permanent damage.

SEL rates are expected to decrease as the supply voltage in modern technology processes is reduced [52].

Single-Event Functional Interrupt

In the case where the event affects a critical signal of the circuit such as a clock or a reset or a critical memory component such as a control register, this abnormal behavior is known as Single-Event Functional Interrupt (SEFI) [39, 40]. SEFI can lead to severe failures of a system although it is not by nature destructive and usually attacks devices of high circuit complexity. While SEFIs were encountered in RAMs, flash memories, FPGAs and high-end microprocessors, other types of circuits may be affected as the designs tend to become more and more complex. A recovery from a SEFI involves refreshing of the corrupted data, reloading the altered configuration and, in some cases, power cycling of the system. Mitigation techniques for SEFIs are usually at the system level and involve online identification of configuration discrepancies coupled with maintenance of restore points in the system to facilitate abnormal operation.

2.4 Fault Models for Modern Technologies

The generic fault modeling approaches described in Sect. 2 can be applied to model defects arising in modern technologies, such as Field-Programmable Gate Arrays (FPGAs) and Network-on-Chips (NOCs). However, due to their well-defined characteristics the usage of the existing fault models has been adapted to the corresponding architectures. Accordingly, the proposed mitigation or protection techniques are taking advantage of their structured design. In this section we discuss the fault model usage and briefly comment how their variations have contributed in coping with dependability threads.

2.4.1 Fault Models for Field Programmable Gate Arrays (FPGAs)

FPGAs are the main representatives of reconfigurable integrated circuits. In the past few years, FPGAs have received huge attention and, hence, seen important development in terms of capacity and speed. With more than a million configurable logic blocks, embedded specialized modules such as digital signal processing slices and high-speed communication modules, on-chip memory (among many other features) [53, 54], FPGAs have found a huge range of applications beyond ASICs prototyping. From industrial control to scientific instrumentation and from automotive to consumers electronics, even critical applications such as avionics and medical electronics have render FPGAs market presence significant. The latter combined with their increased complexity have increased the dependability threads and, consequently, made the development of advanced mitigation techniques a necessity.

FPGA testing approaches can be separated in *post-manufacturing* and *in-field*. Post-manufacturing testing has been widely accepted with the term application-independent testing while in-field refers to post-configuration testing known as application-dependent testing and is extended to Build-In-Self-Testing (BIST) schemes to guarantee the dependability of the device throughout its lifetime [55]. In both cases, the corresponding techniques separately examine the three basic components of the FPGAs, i.e., the logic blocks, the memory cells and the interconnection fabric. In general, FPGA testing considers models for permanent and transient faults that are similar to those described in Sects. 2.2 and 2.3. Despite the modeling match, fault lists for FPGAs are of different nature. Due to the programming of the device, some faults may become redundant either because some blocks/modules of the FPGAs are not used or because of the increased logic redundancy induced by the versatile functionality of the FPGA's cells. In the following paragraphs we discuss how fault models are affected by the specific characteristics of the FPGA structure.

Application-Dependent Versus Application-Independent Dependability

Right after manufacturing it is necessary to perform thorough testing for all possible faults under the considered models. Proposed techniques for this *Application-Independent testing* (AIT) need to exercise many different programmings of the device, known as *configurations*, in order to be able to test all parts of the FPGA (as well as all part combinations) [56–61]. Under each different configuration, standard testing procedures involving fault activation and propagation to observable points are considered, yet the fault list is constructed only for the part of the device employed. Hence, the main target here is to minimize the number of configurations necessary to achieve the desired fault coverage. Note that the configuration of a device is done in a serial fashion and, hence, is much more time consuming than the application of a test. If the device fails for a test in one of the configurations considered, it may still reach the market allowing only configurations for which the applied tests have not detected any faults.

On the other hand, *Application-Dependent* (ADT) approaches consider only the part of the device employed by the configuration of the intended application [55, 62–66]. The application designer should, therefore, in-field develop, together with their design, the test stimuli to achieve the desired fault coverage. Thus, the fault list considers only the faults enabled by the specific configuration. The test stimuli can be either applied once (after configuration) or be embedded in the unused part of the device in order to be activated on-demand by the end-user in an online or offline mode.

On the other hand, *Application-Dependent* (ADT) approaches consider only the part of the device employed by the configuration of the intended application [55, 62–66]. The application designer should, therefore, in-field develop, together with their design, the test stimuli to achieve the desired fault coverage. Thus, the fault list considers only the faults enabled by the specific configuration. The test stimuli can be either applied once (after configuration) or be embedded in the unused part of the device in order to be activated on-demand by the end-user in an online or offline mode.

Fault Models Usage in FPGA Components

As with not programmable devices, FPGA dependability approaches have extensively used the stuck-at fault model, the stuck-open/stuck-short model and the bridging fault model for interconnections, usually in a pairwise fashion [57, 58, 62, 63, 65]. In many cases, more abstracted functional models have been used for modules such as multiplexers and decoders as well as for modules that the actual implementation is not available [62, 67]. In addition, delay fault models have been used, mainly the Path Delay Fault model and the Transitions Fault model [66, 68, 69].

This paragraph indicates how these models have been used in the literature for the three main components of the FPGA, namely logic blocks, memory cells, and the interconnection fabric.

Faults for Logic Blocks

Configurable Logic Blocks (CLBs) are the main component implementing logic in an FPGA. Oriented in a two-dimensional array they consist of one or more Look-Up Tables (LUTs), multiplexers, flip-flops, and possibly some additional functional units such as full adders. The stuck-at fault model is used for all components' interconnections within the block as well as for all components for which the internal structure is available.

For some components that the actual implementation is not available during testing, especially for ADT approaches, and, hence, abstracted functional models are used accordingly. For example, in the case of a standard 2-to-4 decoder the faults are considered as described in Sect. 2.2 under functional fault models.

Another model specifically used for the LUTs of CLBs is the cell fault model [70]. A cell fault describes a mismatch between specific value combination of the inputs and the expected outputs. LUTs configured as RAMs are tested considering memory specific models such as data retention, coupling, and address decoder faults over and above stuck-at and transition fault models [71]. Specifically for the sequential elements, namely the flip-flops, latches, and registers of CLBs the stuck-at and transition fault models are used [72]. The work in [62] combines all these fault models to create a comprehensive fault list for generic FPGAs and this combination is referred as *hybrid fault model*. The work in [63] proposes an extension of the bridging fault model that considers a feedback path where an output of an LUT dominates the one of its inputs creating unexpected behavior of sequential nature.

The mitigation techniques for CLBs usually keep the CLBs under test configuration unchanged and appropriately modify the configuration of the interconnections between them as well as the configuration of the CLBs not used in order to make the CLBs under test controllable and observable. The configurations are loaded in an iterative manner until all faults considered are covered [57, 73–75].

Faults for the Interconnection Fabric

The great majority of the transistors in an FPGA is used for the interconnection of the CLBs. Due to the programmable nature of the device a huge number of possible connections is available and appropriate routing logic is used. The two predominant approaches use either programmable switch matrices or programmable multiplexers. For the routing logic the stuck-at fault model is used, while for the switch matrices the stuck-on/stuck-off variation of the transistor model is considered. For the actual interconnections bridging faults are extensively used, usually in pairwise combination, followed by stuck-open and stuck-short faults [58, 59, 61, 63, 64, 76].

The mitigation approaches keep the interconnection fabric configuration unchanged and appropriately configures the CLBs to activate the desired connection structures in order to test them. For this purpose, a functional model has been proposed to enforce the CLBs to an easy-to-control configuration. In [76] the single-term function is introduced were each LUT is configured to implement either a single minterm or a single maxterm function. Thus, the expected value of the LUT is fixed for all combinations of inputs except one. This configuration of the CLB can be used to activate specific faults in the interconnection fabric by bringing specific value to a desired line driven by the CLB.

Faults for Memory Cells

Based on the process technology followed during manufacturing, FPGAs may follow different configuration philosophies. The most commonly used processes are the SRAM-based and the flash-based FPGAs. In the SRAM-based case the configuration should be loaded from an external source, in contrast to the flash-based. If this external source is a non-volatile memory then it has to be considered for dependability threads as well.

In all cases of memory structures the standard memory fault models are used including pattern sensitive and adjacent cell coupling, address decoding faults and data retention. Moreover, non-permanent faults rate has been increased mainly due to the existence of SEUs and affect the dependability of the device during its entire lifetime [77–82]. Recently, a number of techniques have been proposed considering multiple-bit upset faults as well [83–85].

Mitigation techniques rely on using sequences of systematic reads and writes to the memory structure to detect undesired behavior. Such an example is the popular approach of March tests [71] which has been extended to FPGAs to allow repair action in a technique which is known as Scrubbing [86, 87]. Moreover, a number of embedded techniques have been proposed that work in the fault tolerance direction using techniques such as duplication and comparison or triple modular redundancy [88–93] or following design approaches based on the considered architecture for soft error resilience [94–99].

2.4.2 Fault Models for Network on Chip

The continuous shrinking of the technology feature size allowed fitting billion of transistors in a single chip, that gave rise to an efficient architecture featuring many *intellectual property cores* (IPs) connected together on the same microchip. The idea behind *Chip Multi-Processor* (CMP) architectures is to use the processing power in a distributed manner to accommodate the given workload. The evolution of CMPs has shown that the communication between the various IPs cannot continue to have the traditional form of a single communication bus as this creates two problems: (i) needs global synchronization that delimits the potential performance of the device and the diversity of the various processing units, and (ii) is not scalable to a large number of IPs [100–102]. For these reasons the trend is to follow the successful example of communication networks to create a communication fabric between IPs known as Network On-Chip (NoCs). The NoC concept examines all the aspects of the IP interconnection including topology, component design, architecture, communication protocols, routing and, naturally, its fault tolerance.

Dependability Concerns in NoCs

On the one hand, NoCs should provide the intended communication between the IPs in a reliable manner and, on the other hand, this communication should be done within stringent design constrains imposed by the area overhead, the performance and the energy consumption. Hence, the successful techniques used in communication networks cannot be employed without being adjusted to the NoC environment. The most important sources of threads come from the small sizes of the components and the small logic swings experienced in NoCs. These two factors increase error rates due to the susceptibility of the NoC to electrical noise mainly as a consequence of crosstalk, electromagnetic interference, thermal stressing, and radiation-induced charges [100]. These are on-top of the post-manufacturing defect sources such as process variation [103] and aging effects discussed in Sects. 2.2 and 3, respectively.

Fault Considerations for IPs cores in NoCs

While assessing the dependability of the IPs using traditional test generation techniques, the NoC is used as the medium to deliver the input patterns to IPs, usually referred to as the *Test Access Medium* (TAM). It is common practice that the IP provider also provides input stimuli to the CMP designer to ensure sufficient coverage for the major fault models such as stuck-at, bridging and transition faults [104]. Naturally, dependability approaches are not as straight forward as in monolithic circuits as the input stimuli should reach the IP from the CMP pins (or the scan infrastructure if available). At the same time the IP under assessment should be isolated so that it does not interact with other cores either inactive or assessed in parallel. Hence, the IPs must be surrounded by a standardized logic adapter known as *test wrapper* to direct the TAM’s stimuli to the actual inputs of the IP and obtain the responses from its actual outputs. Many techniques have been proposed to include built-in self test logic in IP cores or use neighboring cores (not in assessment mode) as input stimuli generator or output response analyzer [105–109].

Fault Models for the NoCs Routing Logic

The main components of the interconnection fabric should be also assessed both individual and in combination. Other than the *communication links* (communication channels), the NoC consists of *routers* to carry out the transmission of the data packets at each node, *network interfaces* that provide an access channel from the IP to the NoC infrastructure, and, hence communication with other cores. Although most of the proposed methodologies do not examine the assessment of the communication links separately, the corresponding fault models are of different nature and are, thus, discussed in the next section.

As far as the network interface is concern, the vast majority of related methodologies consider that it is assessed implicitly together with the IP since all input stimuli should pass through it to reach the IP. Usually the network interface is placed within the wrapper circuit. Other techniques are considering them separately or as part of the router assessment; yet they assume traditional design-for-testability (DfT) infrastructure such as scan chains and consider both the stuck-at and transition delay fault models [110, 111]. The work of [112] proposed a functional fault model to exercise all the functional modes of the network interface which serves as the source and the destination for data used to assess the router.

For the router, each of its main components should be explicitly considered. The crossbar switch, the arbiter, and the handshake logic as well as the interconnections between them are usually assessed considering the stuck-at fault model, functional models specific to each component, the stuck open/short model as well as delay fault models [110, 113]. In addition, for the FIFO input buffers memory fault models are also considered [114]. The related methodologies usually take advantage of the canonical form of the network topology to check the routers in an iterative or progressive manner using identical input stimuli [114]. The work of [115] proposed a high level fault model called **link fault model** where a path between two router ports is considered as faulty if it the outgoing data is not identical to the incoming. Similarly, [116] proposes a fault model where the incoming data is forwarded to an incorrect router channel resulting in either lost packages or performance reduction.

Fault Models for NoCs Communication Links

The final component of the NoC infrastructure not considered in the previous sections is the interconnections as such, i.e., the wires. Since many of the interconnections in a microchip are within few nanometers, the natural consequence is to exhibit increased crosstalk levels. On the one hand, the traditional assessment issues impose offline approaches both for post-manufacturing and built-in self test purposes. On the other hand, the increased error rates due to the susceptibility of deep-submicron technology to radiation-induced faults, place online approaches on the top of dependability requirements.

Several online approaches have been proposed adopting techniques from the telecommunication discipline [117, 118]. Error Correction/Detection Codes (ECDC) have been developed within the stringent design budget that provide communication of high reliability using mainly information and time redundancy and less often spatial redundancy [118, 119]. Information redundancy considers techniques such as cyclic redundancy check, hamming encoding and parity checks and is usually employed for error detection. The desired reliability is then achieved by employing error correction either using advance information redundancy approaches or using time redundancy, e.g., retransmitting incorrect packages. According to the study carried out in [117], the latter combination imposes smaller overhead in terms of area overhead and energy efficiency.

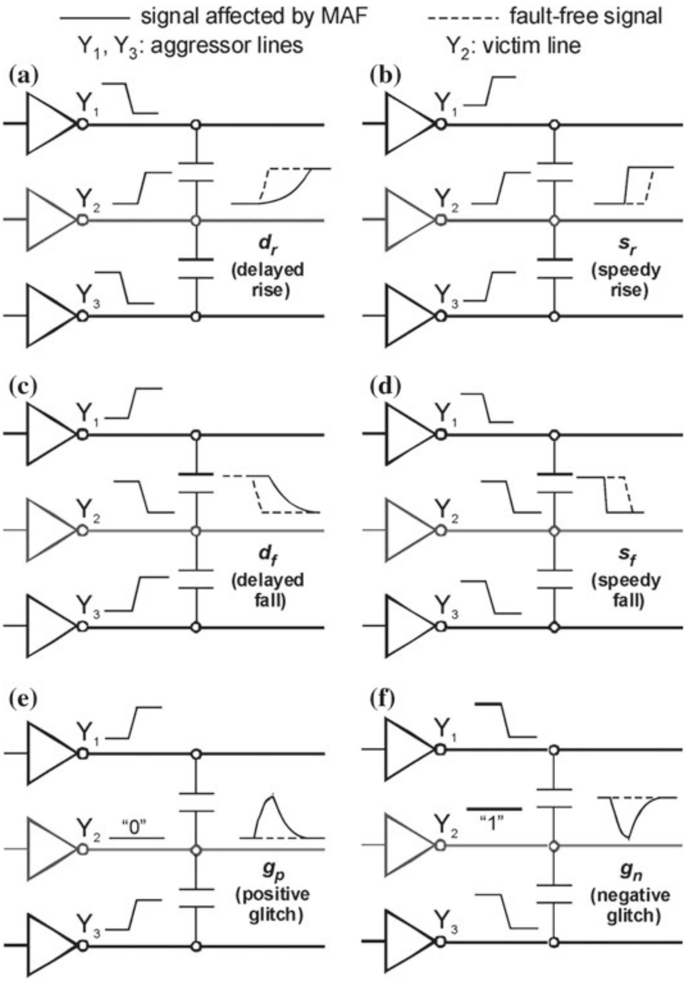


Fig. 5 Crosstalk faults modeled using the maximum aggressor fault model as presented in [120]

Offline techniques usually rely on the DfT infrastructure and cover the entire range of dependability assessment from design verification to BIST. These methods consider the traditional fault models for permanent faults, mainly the stuck-at, stuck-short/stuck-open, delay and in a large extend the bridging fault model. The same models are considered for the evaluation of the online methods. Obviously, bridging faults are more suitable for modeling the main source of defects in the interconnection links, i.e., crosstalk, yet they cannot model delay faults which are very usual in this context. For this reason the technique of [120] considers a model for crosstalk faults known as the **Maximum Aggressor Fault** (MAF) model. In the MAF model the logic value or transition in an interconnection can be affected by neighbor interconnections (aggressors) by delaying or speeding a transition or trigger voltage glitches. Figure 5 obtained from [120] illustrates the six possible faults according to the MAF model. The work in [121] proposed a more generic model for crosstalk where the value of an interconnection can be affected by a neighbor one either by dominating it or by realizing an undesired logical operation (AND–OR) between the two. The fault considers lines in the same neighborhood both in the same or in different metal layers.

3 Aging and Lifetime Reliability

The definition of the aging and wear-out models represents a fundamental step towards the estimation of the lifetime of a digital system and, eventually, to design solutions in order to extend it, should it be not satisfactory with respect to the system mission time. Both academia and industry have broadly investigated such aging models proposing solutions acting at the different abstraction levels (e.g., [122–124]). Because design activities typically migrate to system level to dominate the complexity of the working scenario, aging models have also been abstracted from the device to the system level. Indeed when considering a modern multi-processor system integrating a large number of cores, the estimation of the expected lifetime of the system becomes a challenging activity that can be addressed only at system level, especially when considering that the workload may change over time, thus causing a varying stress and aging on the various components, also due to subsequent processors' failures that force a re-distribution of the workload for the system to continue working.

In this section, we will present a state-of-the-art methodological framework for the definition of aging models and the estimation of the lifetime reliability of a complex digital system composed of many components, such as a multiprocessor architecture. In particular, we will discuss how to compute the reliability $R_{\mathrm{sys}}(t)$ of the system, i.e., the probability of the system to be operational until time t, and, consequently, to derive the related Mean Time To Failure (MTTF). The overall organization of the framework is presented in Fig. 6; the flow is an adaptation of the one in [122, 124] and integrates various features and details of the aging models

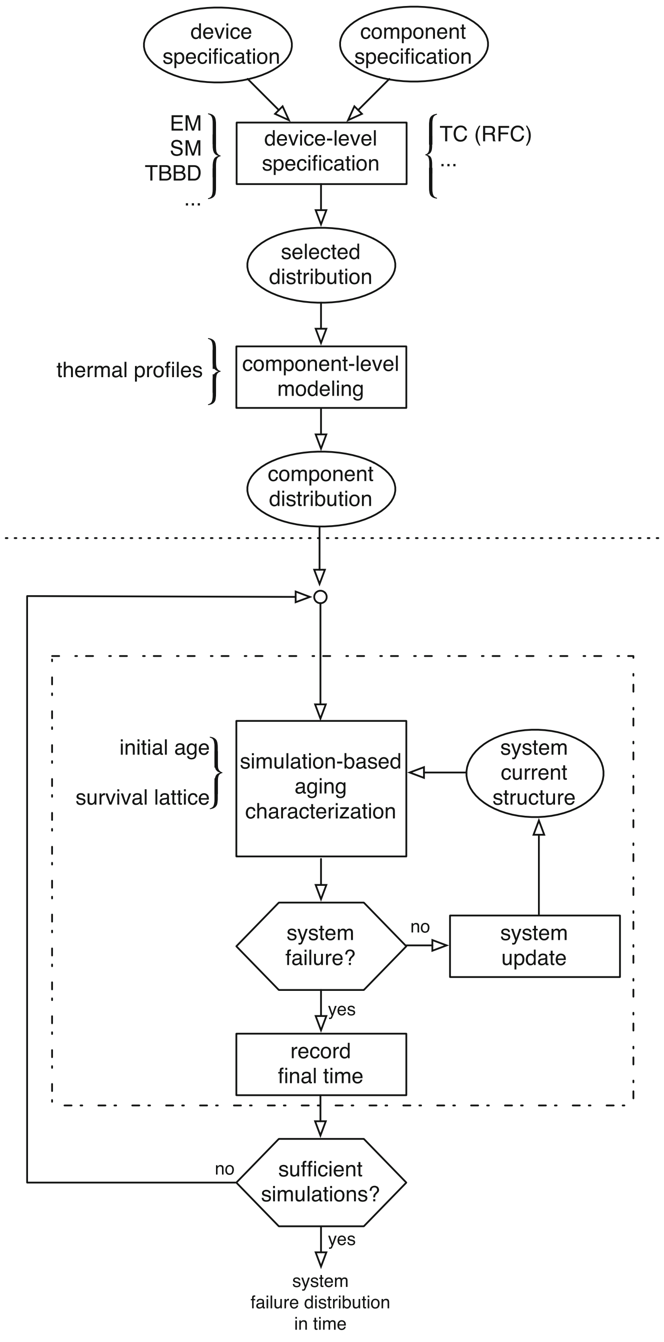


Fig. 6 Reliability modeling and analysis framework

defined in [125–127]. The starting point is the reliability characterization of the single device by means of accelerated experimental campaigns (Sects. 3.1 and 3.2). Industrial standards define how to empirically derive the failure rate related to each failure mechanism in a given class of devices by means of accelerated experimental sessions, and, subsequently, how to identify the statistical distribution that fits best the sample data.

The device level characterization is generally performed in fixed working conditions for sake of simplicity. Therefore, it will introduce a considerable error in the evaluation of a real system used to work in varying working conditions. Thus, a component-level model is defined to incorporate in the $R(t)$ formula the possible variations in the working conditions, and, in particular in the temperature, in a single component (Sect. 3.3).

Finally, a system-level reliability model is defined on the basis of the previous component-level models to take into account all the peculiarities of a multi-component architecture (i.e., the variable workload and the possibility to survive after the first sequences of failures). As discussed in Sect. 3.4, the analytical formulation of such model becomes unmanageable. For this reason a Monte Carlo-based simulation approach is generally adopted. The approach is based on the definition of a *survive lattice*, composed of all the possible working configurations the system may assume, depending on the sequence of occurred failures. Therefore, each simulation will be performed by generating a random sequence of failures on the basis of the component failure distribution. At each failure, the working conditions of the various components will be adapted according to the new system configuration. Then the simulation will proceed until the working configuration is within the survive lattice. At the end, the mean time to failure of the overall system, and the related reliability curve $R_{\mathrm{sys}}(t)$ will be computed by aggregating the actual lifetimes collected with a large number of independent simulations.

3.1 Device-Level Failure Mechanisms

The definition of failure mechanisms for physical devices has been accurately addressed by the electronics industries during the past decades. In particular, their main necessity has been to conduct testing of new technologies and fabricated devices in order to predict failure behaviors during the nominal operational life of the components in post-production phases. Therefore, JEDEC Solid State Technology Division [128], an independent semiconductor engineering trade organization and standardization body that involves some of the world's largest computer companies, defined a standard for the specification of models for each specific physical failure mechanism in electronic components in order to perform testing experiments at accelerated conditions capable of predicting the aging and wear-out behavior of the devices at customer use conditions.

The basic model defined in [123] for a single failure mechanism is based on the Arrhenius equation. This mathematical expression is applicable to most thermal acceleration for semiconductor device failure mechanisms, such as electromigration, time-dependent dielectric breakdown, or thermal cycling, and characterizes the failure rate of the considered class of components or devices at specific working conditions, and in particular considering a fixed temperature. The Arrhenius equations is specified as follows:

$$\lambda = A_0 e^{-\frac{E_a}{kT}} \tag{1}$$

where λ is the failure rate of the considered failure mechanism, A_0 is an experimentally estimated fitting constant, E_a is the activation energy (in eV, depending to the specific failure mechanism), k is Boltzmann constant (equal to $8.62 \times 10^{-5}\,\mathrm{eV/K}$), and T is the steady-state (worst-case) operating temperature (in Kelvin degrees). For simplicity, in the experimental data analysis, JEDEC publication considers an exponential distribution of the failures. Therefore, the Mean Time To Failure (MTTF) of the considered class of devices can be computed as the opposite of the failure rate:

$$\mathrm{MTTF} = \frac{1}{\lambda} = A_0^{-1} \mathrm{e}^{\frac{E_a}{kT}} \tag{2}$$

It is worth mentioning that the adoption of an exponential distribution for modeling aging and wear-out mechanisms is clearly inaccurate since such model has a constant failure rate over time. At the opposite, if we do not consider the infant deaths that are mainly caused by production defects and are screened in post-production testing, wear-out mechanisms typically present a very low failure rate at the beginning of the component lifetime; then, λ increases as the component ages. Therefore, a more adherent model should consider more advanced probability distributions, such as the Weibull or the lognormal, typically adopted instead of the exponential one, and discussed later on.

In post-production testing activities, accelerated experiments are conducted in a high temperature environment to empirically measure the failure rate and, consequently, the mean time to failure of the considered class of devices in stress conditions, called $\mathrm{MTTF}_{\mathrm{test}}$. A large set of devices are considered in these experimental campaigns to have a reasonable statistical confidence of the empirical result. Then, the following rewriting of Eq. 2 is used to calculate the thermal acceleration factor A_T for device MTTF distribution:

$$A_T = e^{\left(\frac{E_a}{k}\right) \cdot \left(\frac{1}{T_{\mathrm{test}}} - \frac{1}{T_{\mathrm{op}}}\right)} \tag{3}$$

where the new symbols T_{test} and T_{op} represent the temperature of the accelerated experiment and the operational temperature in nominal conditions, respectively. Finally, the mean time to failure in nominal conditions $\mathrm{MTTF}_{\mathrm{op}}$ can be estimated as:

$$\mathrm{MTTF}_{\mathrm{op}} = A_T \cdot \mathrm{MTTF}_{\mathrm{test}} \tag{4}$$

The basic model represented in Eq. (2) has been specialized for each failure mechanism, in particular, by specifying A_0 and E_a parameters as a function of other more detailed factors. The specific models for some of the most common failure mechanisms will be discussed in details in the rest of the section. One of the most important issues in all these models is the fact that the actual values of most of the involved parameters and factors are not fixed but need to be experimentally characterized by means of specific empirical campaigns. Another JEDEC publication [123] accurately specified how to conduct such process.

3.1.1 Electromigration

Electromigration (EM) is a phenomenon which occurs in wires and vias as a result of the momentum transfer from electrons to ions that construct the interconnect lattice and leads to hard failures such as opens and shorts in metal lines. The Mean Time To Failure due to EM is given by the Black's model represented by the following equation [123, 129]:

$$\mathrm{MTTF}_{\mathrm{EM}} = \frac{A_{\mathrm{EM}}}{(J - J_{\mathrm{crit}})^n} e^{\frac{E_{a_{\mathrm{EM}}}}{kT}} \tag{5}$$

where A_{EM} is a constant determined by the physical characteristics of the metal interconnect, J is the current density and J_{crit} the threshold current density, $E_{a_{\mathrm{EM}}}$ is the activation energy for EM, k is Boltzmann constant, n is a material-dependent constant empirically determined, and T is the temperature.

The current density J for an interconnect can be modeled as follows:

$$J = \frac{CV_{\mathrm{dd}}fp}{WH} \tag{6}$$

where C, W, and H are the capacitance, width, and thickness of the line, respectively, while f is the clock frequency and V_{dd} the supply voltage. Therefore, it may be noted how electromigration mechanism depends secondarily also on the current flowing the interconnect.

3.1.2 Time-Dependent Dielectric Breakdown

Time-dependent dielectric breakdown (TDDB) is an effect related to the deterioration of the gate oxide layer. In particular, due to hot electrons, gate current causes defects in the oxide, which eventually form a low-impedance path and cause the transistor to permanently fail. This phenomenon is strongly affected by the temperature and the electric field; it increases with the reduction of the gate oxide dielectric layer thickness and non-ideal supply voltage reduction. The MTTF related to this effect is given by the following equation [130]:

$$\mathrm{MTTF}_{\mathrm{TDDB}} = A_{\mathrm{TDDB}} \left(\frac{1}{V}\right)^{(a-bT)} \mathrm{e}^{\frac{A+B/T+CT}{kT}} \tag{7}$$

where A_{TDDB} is a fitting constant, V is the supply voltage, a, b, A, B, and C are empirical fitting parameters. It is worth mentioning that a slightly alternative model is proposed in [123].

3.1.3 Stress Migration

Stress migration (SM) is very similar to electromigration and is a phenomenon where the motion of metal atoms in the interconnects migrate because of mechanical stress, induced by differing thermal expansion rates of the different materials in the device and by the distortions in the crystal lattice of the semiconductor substrate. The model for computing the MTTF is based on thermo-mechanical stresses and is expressed by the following equation [123, 130]:

$$\mathrm{MTTF}_{\mathrm{SM}} = A_{\mathrm{SM}} |T_0 - T|^{(-n)} e^{\frac{Ea_{\mathrm{SM}}}{kT}} \tag{8}$$

where A_{SM} is a fitting constant, T_0 is the metal deposition temperature during fabrication, T is the temperature of the metal layer at runtime, n is an empirically derived constant, and $E_{a_{\mathrm{SM}}}$ is the activation energy for SM.

3.1.4 Thermal Cycling

Thermal cycling (TC) refers to a wear out effect caused by thermal stress due to mismatched coefficients of thermal expansion for adjacent material layers. At runtime, temperature variation produces an inelastic deformation, eventually leading to failure [131]. Rather than estimating the MTTF, usually the number of cycles leading to the failure is computed, by using a modified Coffin–Mason equation [132]:

$$N_{\mathrm{TC}} = A_{\mathrm{TC}} (\delta T - T_{\mathrm{th}})^{(-b)} \mathrm{e}^{\frac{Ea_{\mathrm{TC}}}{kT_{\mathrm{Max}}}} \tag{9}$$

where A_{TC} is an empirically determined fitting constant, δT is the thermal cycle amplitude, T_{th} is the temperature where the inelastic deformation begins, b is the Coffin–Mason exponent constant, $E_{a_{\mathrm{TC}}}$ is the activation energy for TC, and T_{Max} is the maximum temperature during the cycle.

3.1.5 Sum of Failure Rates

In the reliability analysis of a system, it is generally necessary to take into account various failure mechanisms that have been demonstrated to be the most predominant for the considered family of devices or technology. Therefore, JEDEC [123] proposes to adopt the Sum-Of-Failure-Rates (SOFR) approach to combine their contributions and obtain a single failure MTTF in output.

The model is based on two main assumptions: (1) the first occurrence of a failure due to any mechanism will cause the failure of the system, and (2) failure mechanisms are independent of one another. In such a scenario, the overall failure model can be modeled as a series system. Therefore, as discussed in [130] when considering the adoption of the exponential distribution, the aggregated MTTF will be computed as follows:

$$\mathrm{MTTF}_{\mathrm{SOFR}} = \frac{1}{\sum \lambda_i} \tag{10}$$

being λ_i the failure rates associated with the different failure mechanisms. On the other hand, more complex failure distributions will require a more advanced analysis that will be discussed in the following part of the chapter.

3.2 Failure Distributions

The empirically determined equations presented in the previous sections are used to characterize the estimated mean lifetime of a class of single devices. Such equations can be adopted when considering a system constituted by components all in the same working conditions, that never change during their operational life. However, this situation seldom occurs in complex system, such as multi-core ones, constituted by several processing cores, working in different conditions, and usually undergoing dynamic changes in loads and therefore in temperature (and consequently aging effects). In these scenarios is thus necessary to extend the model of the components to consider varying working conditions (i.e., temperature) over time, and to combine the contributions of the various components within the entire system in order to model and analyse the overall system lifetime reliability value. To this end, the first step is to introduce the means to perform an analysis of the failure distribution over the time for a given component, being a functional unit, a core, or any other module considered as an atomic entity. Then, in the subsequent sections, we will extend the analysis to a single component subject to temperature variations (Sect. 3.3), and, at the end, to the overall system constituted by several components (Sect. 3.4).

A classically used time distribution for modeling semiconductor component failures is depicted by the bathtub curve [133], shown in Fig. 7, that represents the failure rates experienced by the class of components over the time. It is possible to

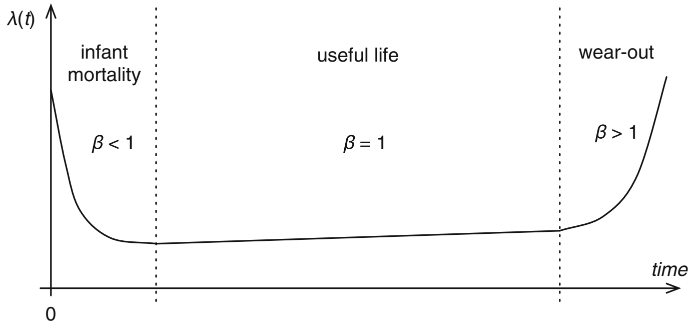


Fig. 7 Bathtub curve

identify three zones. The first zone shows the probability of a component to fail that decreases over time, and these failures are usually due to production defects or packaging problems that arise early in the device life, and it is called infant mortality. The second zone represents the intrinsic or useful life of the component, affected by random failures, mainly not affected by time, with a relatively constant failure rate, that characterizes most of the life of the component. The final zone models wear-out, where the failure rate increases with the age of the component itself. Actually, in aging-related studies, infant mortalities are generally not considered. As a matter of fact, these failures are generally caused by defects introduced during the manufacturing process; moreover, many of these component defects can be removed by effective reliability screens.

In the past decades, the exponential distribution has been commonly adopted, to represent the time distribution in component failures. More precisely, such distribution models the reliability of a component at time t (generally expressed in hours) working at a fixed temperature T as follows:

$$R(t) = e^{-\frac{t}{\alpha(T)}} \tag{11}$$

where $\alpha(T)$ is the scale parameter that is equal to the MTTF of the corresponding failure mechanism modeled above at the fixed temperature T. Indeed, the exponential distribution presents a constant failure rate, and therefore it is capable of approximating the second phase of the curve which generally represents the service lifetime. However, the aggressive trend in technology downscaling has caused an acceleration in the aging and wear-out process, thus anticipating the device failures [123, 124]. For this reason, also in the useful life phase, the failure rate is not constant but at the opposite presents an increasing trend. For this reason, in the last decade the most commonly adopted distributions to model the reliability of a device are the Weibull and the lognormal ones, since their failure rate can be modulated in particular to present an increasing failure rate [123, 124].

The Weibull distribution models the reliability of a single component at a given time t working in stationary conditions, and, in particular, at a fixed steady-state temperature T as:

$$R(t) = e^{-\left(\frac{t}{\alpha(T)}\right)^{\beta}} \tag{12}$$

where β is the Weibull slope parameter (considered to be independent of the temperature), and $\alpha(T)$ the scale parameter. Moreover, according to the properties of the Weibull distribution and considering the characterizations defined in Sect. 3.1, the scale parameter can be specified for a given failure mechanism as follows:

$$\alpha(T) = \frac{\mathrm{MTTF}_i}{\Gamma\left(1 + \frac{1}{\beta}\right)} \tag{13}$$

being MTTF_i the MTTF equation of the considered failure mechanism (as presented in Sect. 3.1) and Γ the statistical Gamma function. As an example, the slope parameter can be expressed for the electromigration mechanism in terms of its physical parameters as:

$$\alpha(T) = \frac{A_{\mathrm{EM}}(J - J_{\mathrm{crit}})^{-n} e^{\frac{Ea_{\mathrm{EM}}}{kT}}}{\Gamma\left(1 + \frac{1}{\beta}\right)} \tag{14}$$

From this formula it is possible to clearly note the dependence of the reliability of a component from its temperature.

One of the reasons motivating the adoption of the Weibull distribution is the fact that it can represent the entire bathtub curve of a component, by composing three different distributions (with different β values), while the other distributions can only model a part of it (e.g., the exponential distribution fits the useful life, and the lognormal one the wear-out). More precisely, the infant mortality phase of life is characterized by $\beta < 1$, the useful life by $\beta = 1$ (corresponding to an exponential distribution), while the wear-out phase has $\beta > 1$.

Another commonly used statistical distribution to model wear-out related mechanisms is the lognormal distribution, which represents the reliability of a component working at a fixed temperature T at time t as follows:

$$R(t) = \frac{1}{2} - \frac{1}{2}\,\mathrm{erf}\left(\frac{\ln(t) - \mu(T)}{\sqrt{2\sigma^2}}\right) \tag{15}$$

where $\mu(T)$ is the scale parameter, σ the shape parameter and erf the error function. Moreover, similarly to the Weibull distribution, $\mu(T)$, can be computed for a specific failure mechanism with MTTF_i as:

$$\mu(T) = \ln(\mathrm{MTTF}_i) - \frac{\sigma^2}{2} \tag{16}$$

In order to select the most appropriate failure distribution, JEDEC proposes in [134] a method based on specific acceleration experiments and a subsequent fitting of the observed data by means of a regression technique with maximum likelihood. Literature has noted that the lognormal distribution applies to EM failures [135], while the Weibull one to the TBBD phenomenon [136].

3.3 *Modeling Reliability of a Component in Variable Working Conditions*

The reported formulas for the estimation of the MTTF associated with a given physical phenomena can be adopted in a working scenario characterized by a steady-state operation mode, otherwise when temperature variations occur (for instance due to changes in the workload) such models can lead to large errors. As a simplification, a fixed, worst-case constant temperature is adopted to estimate MTTF in the scenario of accelerated testing of a class of devices; however, this introduces significant limitations, when considering an early-stage system-level analysis of multi-core architectures constituted by many processors, not necessarily all working at all times, and used in highly dynamic contexts. Therefore, reliability computation of the single component is here extended to the case of arbitrary temporal temperature variations; in that way, we provide the means to discuss the case of a complex multi-core system in the next section.

As discussed in [122, 137], when the temperature of the component varies over the time, to model its reliability at a given time instant t, it is necessary to consider the current working conditions as well as the aging due to the *sequence* of previous periods of activity, each one characterized by a different temperature. To such purpose, it is possible to model the activity of the component in terms of tuples (t_i, T_i) specifying that in the period between t_{i-1} and t_i the device temperature is T_i (steady-state temperatures are considered for sake of simplicity), such that a series of tuples defines the overall temperature trace from the initial instant up to the current one $[(t_1, T_1)(t_2, T_2)\ldots(t_i, T_i)]$, as shown in Fig. 8.

In this scenario, for t included in $[0, t_1]$, it is possible to use the $R(t, T_1)$ formula of the adopted failure distribution by specifying the temperature T_1 to compute the current reliability value r; in fact at the beginning of the initial step the component

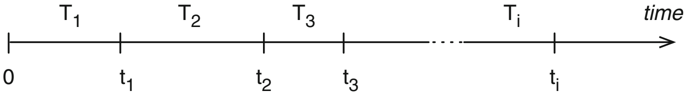


Fig. 8 Model of a component in terms of temperature variations over time

is new and therefore no previous aging status has to be taken into account. Let assume that the system has been running at a temperature T_1 from time 0 to time t_1, and a temperature change occurs, which becomes T_2. For $t > t_1$, the $R(t, T_2)$ formula cannot be directly applied by specifying a different temperature T_2 because such formula would not be able to take into account the aging experienced in the initial period. This situation can be clearly noted in the left plot in Fig. 9. In particular, the two reliability curves related to the two different temperatures have different shape parameters and therefore their slopes differ, causing a discontinuity in the temporal failure probability distribution. To *reconcile* such a discontinuity, it is necessary to right-shift the second curve to connect with the first one at time t_1, as shown in the right side of Fig. 9; in this way, the second curve will take into account the aging accumulated during the first period.

From a mathematical point of view, such a right-shift is performed by manipulating the actual time value to be passed to the $R(t, T)$ formula in the period $[t_1, t_2]$ as follows: $R(t - t_1 + \hat{t}_1, T_2)$, where $\hat{t}_1$ is the time instant satisfying the condition $R(t_1, T_1) = R(\hat{t}_1, T_2)$. To compute $\hat{t}_1$, the inverse of the reliability function $R(t, T)$ is used, i.e., $\hat{t}_1 = R^{-1}(r_1, T_2)$ [137]. As an example, when adopting Weibull distribution, $R^{-1}(r, T)$ corresponds to:

$$R^{-1}(r, T) = \alpha(T) \cdot (-\log(r))^{1/\beta} \tag{17}$$

The discussed right-shift has to be performed at each temperature change on the new curve to connect to the previous one. Moreover, as discussed in [124], by manipulating the formulas within the equation $R(t - t_1 + \hat{t}_1, T_2)$ and by applying it on the overall temperature trace, it is possible to rewrite it in a simpler way as follows:

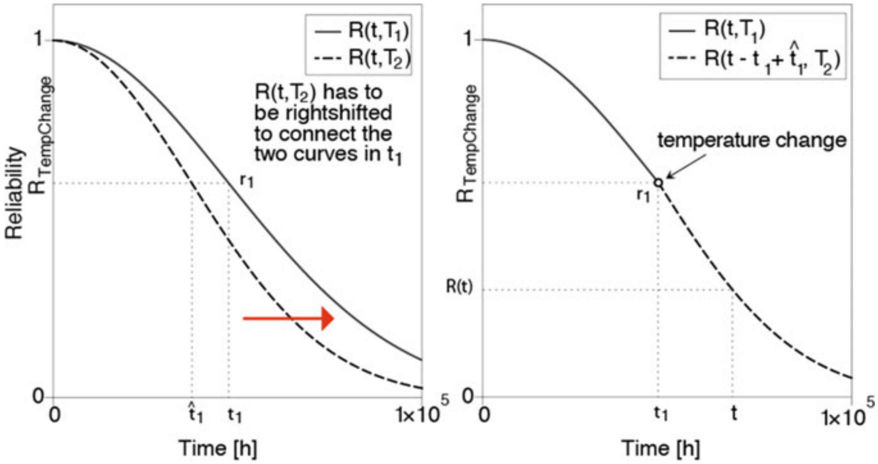


Fig. 9 Reliability curve reconciliation considering temperature changes

$$R(t) = e^{-\left(\sum_{j=1}^{i} \frac{\tau_j}{\alpha(T_j)}\right)^{\beta}} \tag{18}$$

where τ_j represents the duration of each period of time with constant temperature T_j until time t, i.e., according to Fig. 8, $\tau_j = t_j - t_{j-1}$ and, therefore $t = \sum_{j=1}^{i} \tau_i$). Finally, the mean time to failure of the component can be computed as the integral of the area below the reliability curve:

$$\text{MTTF} = \int_0^{\infty} R(t)\mathrm{d}t. \tag{19}$$

by using the $R(t)$ formula obtained by means of the variable temperature profile in Eq. 18. It is worth noting that it is possible to derive the corresponding formula for the lognormal distribution with a similar approach.

3.3.1 Practical Considerations on Reliability Computation

In real situations, Eqs. (18) and (19) cannot be directly applied for the analysis of the lifetime reliability of a class of components for the following reasons. First, the model requires to know the overall temperature trace from 0 to ∞, (that can be approximated to the instant of time where the reliability formulas returns a value very close to zero). Actually, the computation of such a temperature profile would be unmanageable by performing real measures on the device due to the very long experiment duration; to partially mitigate this problem, accelerated experiments may be performed. Nevertheless, when using a simulation framework we may face the same issue, even if, in this case, it is possible to use some strategies to accelerate the simulation in cases where the temperature stays constant for a period. The second issue with the considered model is more theoretical. Actually, the defined model is stochastic; it defines the failure time of a component in a probabilistic way. Therefore, we cannot use a single run to characterize the reliability of a class of components. At the same time, during a single run, the component will fail with a high probability before its reliability is approximately equal to 0.

Therefore, to solve these issues from a practical point of view, in various works [122, 124–126] authors proposed to identify a characteristic temperature profile of a reduced time period (considerably smaller than the overall lifetime) able to be representative of the average behavior of the component for almost-stationary conditions. In particular, when a characteristic temperature profile is collected by means of real measures or simulations, it is possible to compute an average aging rate α_{avg} as:

$$\alpha_{\text{avg}} = \frac{\sum_{i=0}^{p} \tau_i}{\sum_{i=0}^{p} \frac{\tau_i}{\alpha_i(T)}} \tag{20}$$

where τ_i represents the duration of the p steps (each one having a steady-state temperature on the component) within the simulated period having a duration equal to $\sum_{i=0}^{p} \tau_i$. Therefore, such aging rate can be used directly in the original formulation of the reliability curve in Eq. 12 as shown below for the Weibull distribution:

$$R_{\text{Weibull}}(t) = e^{-\left(\frac{\sum_{i=0}^{p} \frac{\tau_i}{\alpha(T_i)}}{\sum_{i=0}^{p} \tau_i} \cdot t\right)^{\beta}} \tag{21}$$

while for the lognormal one, μ_{avg} formula is similarly defined and replaced in Eq. 15, as here shown:

$$R_{\text{lognormal}}(t) = \frac{1}{2} - \frac{1}{2}\,\text{erf}\left(\frac{\ln(t) + \ln\left(\frac{\sum_{i=0}^{p} \frac{\tau_i}{e^{\mu(T_i)}}}{\sum_{i=0}^{p} \tau_i}\right)}{\sqrt{2\sigma^2}}\right) \tag{22}$$

It is worth noting that, as discussed in [126] and empirically proved in [122], when having a characterizing period with $\sum_{i=0}^{p} \tau_i \ll \text{MTTF}$, the error introduced by the approximation is really negligible.

It is worth noting that in many situations, a single characterizing temperature profile may not be sufficient to describe the service life of a component. This situation may occur when the components work in different use modes on the long term period and therefore the approximated method presented above may be not very accurate [122]. In such a situation, it is necessary to characterize each use mode as discussed before. Then, if a typical distribution of the use modes is known in the long term period, the reliability of the component can be computed by using Eq. (18) using the various average aging rates in the different long term periods. Otherwise, when the distribution of the used modes is unpredictable, as an alternative, the approach proposed in [125] supporting a probabilistic distribution of the use modes can be adopted.

3.3.2 Reliability Computation for Thermal Cycling

The formulation proposed in the previous two sections for the analysis of the reliability of a single component presenting a variable temperature profile is based on an analysis of the steady-state levels in the temperature trace. Therefore, it can be employed for all the fault mechanisms discussed in Sect. 3.1 but thermal cycling. In fact, while the formers depend on the actual temperature of the component, the latter depends not only on the maximum temperatures the component experiences but also on the amplitude of temperature variations between contiguous local minimum and maximum peaks, called thermal cycles.

Also when considering thermal cycling, the component may experience temperature variations with different amplitudes in different instants of time. Therefore, similarly to the other failure models, the basic formula in Eq. (9) cannot be directly applied since it assumes all cycles to be almost equal. Moreover, as for the previous cases, for feasibility reasons the study to determine the reliability curve has to be performed on one or a set of representative temperature profiles. In conclusion, the proposed reliability model needs to be extended to identify and analyze the thermal cycles.

A preprocessing phase need to be added to the model to identify the thermal cycles in the representative temperature profile (or a set of traces) The first step of the preprocessing phase is devoted to the extraction of the peak–valley trace from the representative temperature trace. In this phase, only points representing a temperature local minimum and maximum, called valleys and peaks respectively, are maintained, and the new graph is obtained by connecting subsequent peak–valley (and valley–peak) pairs.

Then, the second preprocessing step consists in the identification of the actual thermal cycles. A thermal cycle occurs when the temperature of the components varies from an initial value, representing a local minimum or maximum, to the opposite extreme point, i.e. a local maximum or minimum, and then returns to the starting point. The computation of thermal cycles is not trivial. A naive computation of the cycles in peak–valley graph in Fig. 10 based on a sequential scan of the graph would return V0-P0, V1-P1, and V0-P1. However, from a more careful analysis, V1-P1 and V0-P1 overlaps. Therefore such a naive approach would lead to a double-counting of cycles, and, therefore, to a considerable error in the subsequent reliability computation. For this reason, as suggested in [124], a specify approach, called *rainflow counting algorithm* [138], needs to be adapted to the correct computation of cycles.

Then, as discussed in [124, 127], when considering a single representative temperature profile composed on m cycles, it is possible to compute the number of cycles to failure N_i caused by each thermal cycle i by means of the Coffin–Mason equation (Eq. (9)) and to combine them to compute the approximated mean number of cycles to failure by means of the Miner's rule [139] as follows:

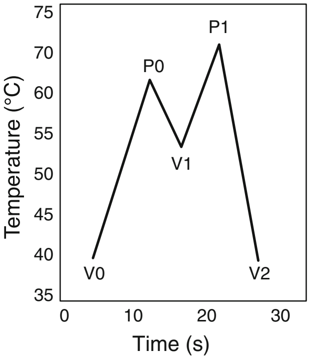


Fig. 10 Example of temperature peak-valley graph

$$N_{\mathrm{TC}} = \frac{m}{\sum_{i=0}^{m} \frac{1}{N_i}} \tag{23}$$

It is worth noting that the formula has been approximated to consider all cycles to have the same duration; such assumption is reasonable because the overall duration of the trace is considerably smaller than the overall lifetime of the component.

Thus, the MTTF related to the thermal cycling can be computed as follows:

$$\mathrm{MTTF}_{\mathrm{TC}} = \frac{N_{\mathrm{TC}} \cdot \Delta t_{\mathrm{trace}}}{m} \tag{24}$$

where $\Delta t_{\mathrm{trace}}$ is the overall duration of the trace.

Given above equations, it is possible to use the reliability model defined in the previous sections also for the thermal cycling failure mechanism. In particular, when considering the Weibull distribution, it is possible to compute the scale parameter α for the reliability curve $R(t)$ by means of Eq. (13). Furthermore, it is also possible to consider several representative temperature traces, by preprocessing them independently and combining them according to the model defined in Eq. (18). Finally, it is possible to use the lognormal distribution in a similar way.

3.4 Modeling Reliability of a Complex Multi-component System

By referring to the above discussion, a model for the single component lifetime reliability computation can be obtained, also considering varying working conditions. We will now extend such a model to a system consisting of several n components (e.g., processors). Different working configurations can be considered, based on the fact that the entire workload is shared between the n processors, or that only a subset of them are actually used, $m < n$, while the others are considered spares to be used once a core fails. Nevertheless, given a workload (that may also change over time) the system is able to perform its mission provided that k-out-of-n processors are have not failed, namely k-out-of-n:G.

In some situations, the system may be composed of several components and to guarantee the correct behavior of the system all of them have to be properly working. This is the common case when we analyze a single processor from an architectural point of view and we consider each functional unit to be a separate component. In this case, components (units) are in a *series* configuration, and the reliability of the entire system can be computed as follows:

$$R_{\mathrm{sys}}(t) = R_1(t) \cdot R_2(t) \cdot \ldots \cdot R_n(t) = \prod_{i=1}^{n} R_i(t) \tag{25}$$

where $R_i(t)$ is the reliability curve of each component i. When a Weibull distribution applies, the above formula can be re-written as:

$$R_{sys}(t) = \prod_{i=1}^{n} e^{-\left(\frac{t}{\alpha(T_i)}\right)^{\beta_i}} = e^{-\sum_{i=1}^{n}\left(\frac{t}{\alpha(T_i)}\right)^{\beta_i}} \tag{26}$$

When considering an architecture consisting of several processors, such as a multi-core one, the workload is typically shared on all processors, or on a subset of them while the rest is used as a spare in case a failure occurs. In this scenario, the entire processor is considered as a single "component", and the system is functional until one processor works properly (provided it can achieve the required performance in executing the workload). As a result a *parallel* configuration is in place, and the reliability of the entire system can be computed as:

$$R_{\mathrm{sys}}(t) = 1 - (1 - R_1(t)) \cdot \ldots \cdot (1 - R_n(t)) = 1 - \prod_{i=1}^{n}(1 - R_i(t)) \tag{27}$$

In this case, when a Weibull distribution applies, the above formula can be re-written as:

$$R_{\mathrm{sys}}(t) = 1 - \prod_{i=1}^{n}\left(1 - e^{-\left(\frac{t}{\alpha(T_i)}\right)^{\beta_i}}\right) \tag{28}$$

In general, Formula (27), accounts for all possible situations that may occur in a system constituted by n components (processors, or cores) when one after the other all cores fail, but one. However, in such an equation, $R(t)$ is more complex that the basic formulation proposed in the previous sections because it is necessary to take into account the fact that (1) any processor may fail with a given failure probability density $f_i(t)$ in any instant of time, (2) the workload may be consequently redistributed on the other processors after a failure, and (3) many different sequence of failures may occur. For this reason, the reliability of the entire system can be specified in an easier way by decomposing the various failure configurations by means of the formula of the total probability:

$$R_{\mathrm{sys}}(t) = P_{\mathrm{no}\_f}(t) + P_{1\_f}(t) + \cdots + P_{k-1\_f}(t) \tag{29}$$

where $P_{\mathrm{no}\_f}(t)$ is the probability that no failure occurred and is computed as the product of the reliability of all processors:

$$P_{\mathrm{no}\_f}(t) = \prod_{i=1}^{n} R_i(t). \tag{30}$$

Then, $P_{1\_f}(t)$ is the probability that at time t there is one failed component. Since n is the number of components, there are n different cases with one component failed at time t, and such an event may have occurred at any time instant t_1 in $[0, t)$. Similarly, $P_{2\_f}(t)$ is the probability that at time t there are two failed components. In this case, there are a number of permutations of having two of the n components failed, with the events occurring at time instants t_1 and t_2. The computation progresses to take into account all possible situations until the number of healthy components is not sufficient to achieve the overall desired system functionality (performance, Quality of Service). Each one of these situations characterized by a number of healthy components and a number of failed ones, together with the load distribution on the healthy ones, represents a point in the so-called *survival lattice* (Fig. 6).

To compute the mentioned probabilities associated with the events it is necessary to refer to the components' failure probability density function $f_i(t)$ of $R_i(t)$ of processor i. More precisely, when considering $P_{1\_f}(t)$, it can be computed as:

$$P_{1\_f}(t) = \sum_{i=1}^{n} \int_0^t f_i(t_1) \cdot \prod_{j=1, j \neq i}^{n} R_j(t|\mathbf{t_1^i}) dt_1 \tag{31}$$

where $f_i(t)$ is the probability density function of $R_i(t)$, i.e., the probability that the failure occurred on processor i at a specific time t_1, and $R_j(t|\mathbf{t_1^i})$ is the conditioned reliability function of processor j knowing that processor i failed at time t_1. Similarly, $P_{2\_f}(t)$ is computed as follows:

$$P_{2\_f}(t) = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \int_0^t \int_0^{t_1} f_i(t_1) \cdot f_j(t_2|\mathbf{t_1^i}) \cdot \prod_{m=1, m \neq i \neq j}^{n} R_m(t|\mathbf{t_1^i}, \mathbf{t_2^j}) dt_1 dt_2 \tag{32}$$

by using probability density functions and the reliability function conditioned by the previous sequence of failures, the first one occurred at t_1, the second one at $t_2 > t_1$. Finally, the general scenario considering the sequence of $k - 1$ failures is a $(k - 1)$-dimensional integral that can be written recursively on the basis of Equations from (31) to (32).

It is worth noting that all contributions depend on the working conditions of the component (the core), that is the workload being executed by the processor, that reaches a certain temperature. Such working conditions may change during the lifetime of the device because the workload is not constant over time, or because—in a multi-core system—the load is re-distributed between the cores, either periodically (according to an aging-aware strategy) or as a consequence of a core failure when its load has to be migrated to the surviving units. Thus, the conditioned reliability functions can be computed by using the formulas for load remapping, as shown in [122].

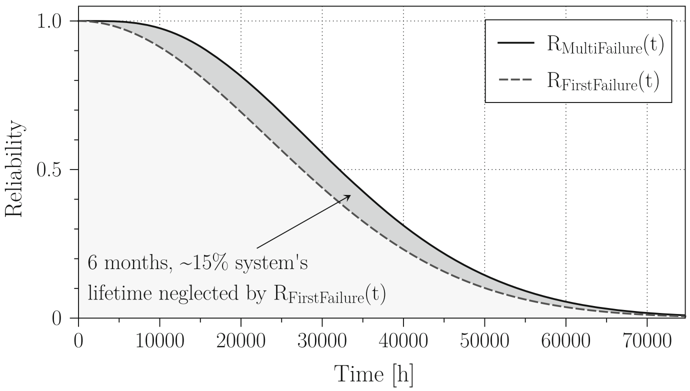


Fig. 11 A remarkable difference in the lifetime estimation can be found even in very simple architectures, as the one considered in this experiment: only two cores executing in parallel and sharing a load. Being the area underlying the curves the MTTF of the system, the *dark grey* area between the *two curves* represents the lifetime portion that is neglected by the *FirstFailure* approach

It is clear that the direct numerical evaluation of a k-out-of-n:G system becomes impractical when the number of k increases, since it combines the complexity of varying working conditions (discussed in Sect. 3.3) to the complexity of multiple components. Indeed, in such situations the computation of $R_{\mathrm{sys}}(t)$ requires the solution of a summation of various multidimensional integrals. As a consequence, some solutions adopt an approximation by considering the end of a multi-core system lifetime the occurrence of the first failure (therefore seeing the system as a series architecture). Indeed, this is a strong approximation for systems exhibiting a redundant number of cores, suitable to run heavy workloads but also designed to survive cores' failures. Figure 11 highlights the difference between the computation of the exact MTTF and the MTTF approximated to the first failure of a system consisting of two cores only. Such a difference further increases as the number of core does.

3.4.1 System Reliability Computation

In literature, it is possible to find various methods for the computation of MTTF in the case of multi-component systems with uneven initial load distribution and re-distribution after components' failures. The most common ones exploit Markov chains, cumulative trapezoidal numerical integration, or Monte Carlo simulations.

Markov chains are a widely used tool for reliability analysis [140], since they usually allow to greatly simplify problems' complexity. However, when coming to the considered problem things become more complicated even for Markov chains; varying fault probabilities and the need for keeping memory of the previous states of the system, force the Markov chain model to drop the memory-less assumptions and adopt the continuous-state model or more complex models. Nonetheless, Markov chains can be used only for exponential failure distributions. In fact, since the other failure distributions (such as the Weibull one) have a memory of the past events, they would require a mixed continuous discrete state space, thus leading to an unfeasible numerical computation.

Then, the second approach, used in [126], is based on the approximation of the formulas to be integrated by means of a set of trapezoids put side by side. Unfortunately, this approach can be employed till two subsequent failures with reasonable execution times.

Methods based on Monte Carlo simulations have proved to overcome these limitations and allow for an efficient, yet accurate, computation of the system lifetime. More precisely, alternative approaches (e.g., [122, 125, 141, 142]) have been devised to cope with the complexity of a direct computation of the lifetime reliability of a system with variable working conditions, and this is even more necessary when considering complex architectures, where computation becomes unfeasible when considering more than 4 subsequent failures. Such approaches are based on simulation frameworks for i) the estimation of the aging rates of the components of a complex systems given a certain workload distribution (to be extracted from the thermal profiles of the components executing a specified load for a window of time), and ii) the computation of the reliability curve $R_{\mathrm{sys}}(t)$ of the entire system given a randomly generated sequence of failures of the system components. This approach is the one shown in the bottom part of Fig. 6 and re-presented in more detail in Fig. 12.

Such solutions are based on the following input information: (i) the system's architecture (number of processors, number k working units for the system to be functional), (ii) the workload, (iii) the failure mechanisms of interest, and (iv) the representative operating configurations, modeled in the survival lattice. The architecture and workload models are necessary to characterize from a thermal point of view the system components; if they can be simulated the computation of the working conditions can be performed online, by means of a simulation engine, otherwise a repository such information can be accessed. The considered failure mechanisms are selected among the ones discussed earlier. The last input is the description of all possible working conditions of the architecture, corresponding to different healthy/failed processors with different workload distributions. These operating configurations are organized in a lattice structure whose size is bounded by $n - k$ failures, according to the designer's specifications. Each lattice element is annotated with a set of mappings, each one describing how the workload is distributed on the healthy processors, information necessary to evaluate the working conditions and consequently extract the thermal profile for the aging computation. When considering a system where changes in the workload distribution are only triggered by a processor's failure, a single mapping is associated with each element in the lattice, while when considering a dynamic workload scenario, the time plan of the various workloads is provided, by listing the set of considered mappings, each one with its own duration.

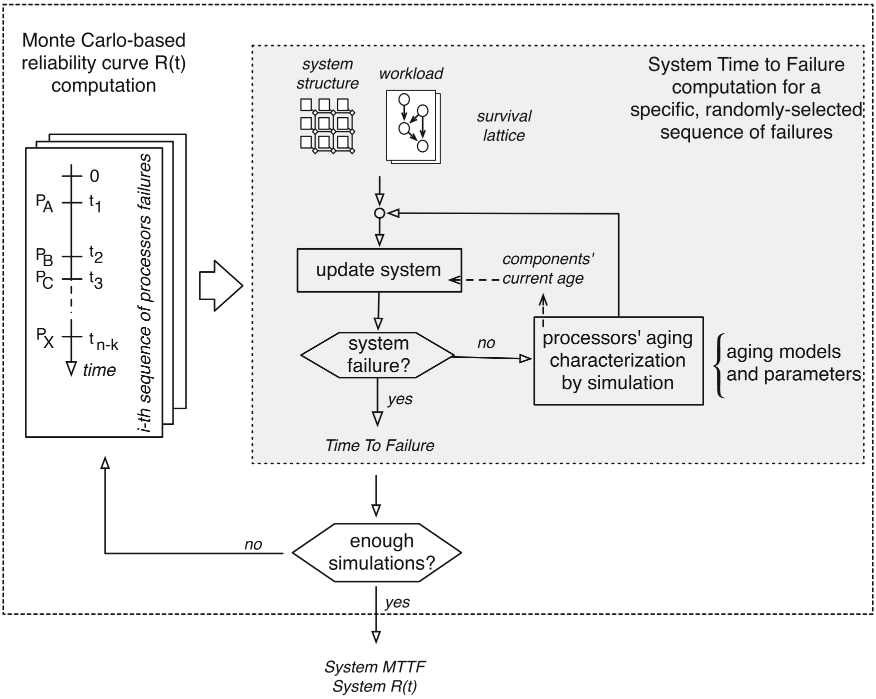


Fig. 12 Simulation framework for computing reliability of a multi-component systems

Within this class of frameworks, simulation engines are exploited to characterize performance, energy consumption and the thermal profile, for all the input configurations, with the aim to compute—as efficiently and accurately as possible—the aging rate for each working condition. Such outcome is then used to evaluate the lifetime reliability of the entire system. For this second activity, a Monte Carlo-based simulation approach is typically adopted to generate a random sequence of failures based on the appropriate failure distribution. Such sequence of failures actually corresponds to a sequence of nodes of the survival lattice; and at each failure the corresponding working conditions of each processor are taken into account, and the system is simulated to compute temperature profiles and aging rates, leading to the computation of the $R(t)$ for that window of time, until the next failure. When the number of fault-free processors does not allow the system to achieve the required performance levels, or—if no requirements have been expressed—when the last processor fails, the process comes to an end and the overall system reliability curve $R_{sys}(t)$ and, based on that, the expected lifetime, expressed in terms of MTTF, can be computed.

A single Monte Carlo test based on a random walk consists of a simulation of a randomly-chosen sequence of processor failures' events; the outcome is the time to failure (TTF) of the overall system during that specific simulation. This process is iterated a number of times to obtain a significant confidence level in the computation of the lifetime reliability, because Monte Carlo simulations are a stochastic approach. Finally, the overall $R_{\text{sys}}(t)$ is computed by aggregating the lifetimes collected from the large set of independent simulations.

3.4.2 An Example of Reliability Computation of a Multi-Core System

In order to give an example of analysis that can be performed with the described Monte Carlo-based method to estimate the reliability of a system composed of different cores, we show here a case study employing the reliability analysis tool defined in [122, 143]. The adopted framework interfaces the Monte Carlo reliability-estimation tool with a functional system-level simulator for workload distribution on multi-core systems within the High Performance Computing (HPC) scenario.

We defined a variant of the problem instance discussed in [143]. In particular, the considered architecture is a homogeneous multi-core system; two different instances, having 3 × 3 and 4 × 4 cores respectively, have been taken into account. Moreover, the system executes workloads modeled in terms of independent jobs presenting various arrival time distributions (Periodic and Poisson ones) and different system load intensities, dubbed as Heavy, equal to the 60% of the system maximum load, and Light ones, equal to 30% (further details on the system characterization can be found in [143]).

The goal of the case study is to evaluate the effect of different workload distribution policies on the overall system's lifetime by considering the fact that the system continues working after a sequence of failure until it is able to guarantee the specified minimum Quality of Service (QoS) level. Three different policies have been considered:

- `uniformAging:` assigns the next job on the idle core having maximum α_i value; in this way it balances the aging among the cores.
- `basicSpare:` uses the minimum subset of the available cores to guarantee QoS while the others are tagged as spare. Then, among the active cores, the next job is assigned to the first available idle core.
- `uniformUsage:` assigns the next job on the idle core having maximum utilization; in this way it balances the utilization among the cores.

Figure 13 presents the output of the Monte Carlo framework. Each scenario considering an architecture instance (3 × 3 or 4 × 4), a workload arrival time distribution (Periodic or Poisson), a workload intensity (Light or Heavy) and workload distribution policy is evaluated and the corresponding bar is added to the chart. The bar is actually stacked and represents the duration of each period between two subsequent MTT k F experienced by the architecture until the complete system failure. As it can be noted, the generated graph allows analyzing the various configurations in terms of maximum number of failures the system is able to tolerate and in the efficiency of each considered workload distribution policy. For instance,

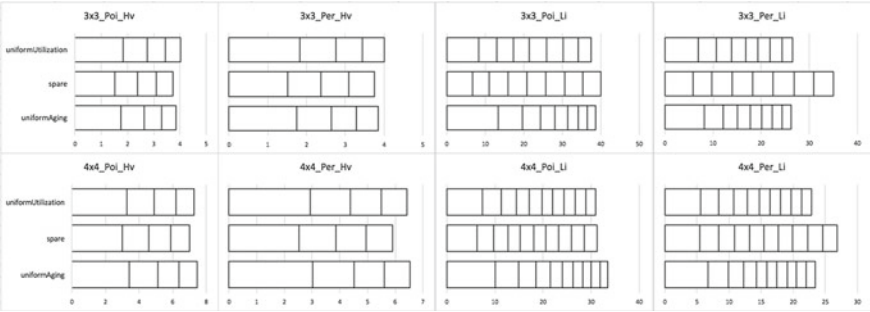


Fig. 13 Empirical analysis of different load distribution policies

in the 3 × 3 architecture, the system is able to guarantee the QoS till to 7 core failures for light workloads and 3 core failures for heavy ones, respectively. In the 4 × 4 architecture, these values reach 10 and 4 core failures, respectively. Moreover, we may notice that there is not a best policy; at the opposite in each working scenario a different policy appears as the most promising (Fig. 13).

4 Metrics

Metrics allow us quantify the effects of a phenomenon or of a proposed solution in a standardized form to assess a certain condition or to aid in decision making and/or process improvement. In this context, we exploit metrics in order to evaluate the dependability of a system (or parts of it) with respect to the possible threats. Broadly speaking there are two types of reliability metrics as described below [144]: (1) Constant Rate Reliability, and (2) Probability of Success Metrics.

The former are commonly referred to as "Mean Life" metrics, and represent a good approximation of the flat region of the reliability bathtub curve. The latter mainly express the probability that a system performs the required function, under stated conditions for a selected period of time, typically exploited when systems do not exhibit a constant failure rate. We here focus on the first class of metrics, to support the analysis of the system to be carried out either at design time, or after production, as well at different abstraction level, based on the adopted models.

Within this class of metrics, we have already mentioned and adopted Mean Time To Failure (MTTF) and Reliability $(R(t))$ in Sect. 3. Mean Time Between Failures (MTBF) is another common metric, suitable for repairable systems, that can be put back in a functional state after undergoing a failure. Another popular metric referring to the failure rate is the Failures in Time (FITs), or failures per billion hours of operation. As a common practice, a constant FIT is adopted, using the relationship $\mathrm{FIT}s = \frac{10^9}{\mathrm{MTTF}}$, suitable only for exponential failure distributions [145], and therefore not appropriate for wear-out based failures, with probability

Table 1 Typical reliability metrics

Metric	Description
Failures in time (FITs)	Failures per billion hours of operation
Mean time to failure (MTTF)	Average time before a failure occurs causing the system to become non-operational
Mean time between failures (MTBF)	Average time between two subsequent failures causing the system to be down until repaired (for repairable systems)
Soft error rate (SER)	Rate at soft errors will affect the system

distributions that change over time. Finally, when considering soft errors, which actually do not cause permanent effects in the system and therefore allow for the system to be considered "repairable", a metric referring to the rate of such events is the Soft Error Rate (SER). These metrics are reported in Table 1. It is worth mentioning that it is possible to apply them at different levels of abstraction, circuit, component, (micro)architectural or system level, in association with the probability function distribution for the fault model of interest. Likewise, metrics can be used either on the nominal system as well as the hardened one, to evaluate the effectiveness of fault detection and mitigation techniques and strategies, and this analysis can be performed both ad design time, or on the prototype, either by means of simulation or experimental campaigns.

References

1. A. Aviziens, Fault-tolerant systems. IEEE Trans. Comput. **12**, 1304–1312 (1976)
2. P.K. Lala, An introduction to logic circuit testing. Synth. Lect. Digit. Circ. Syst. **3**(1), 1–100 (2008)
3. Cristian Constantinescu, Trends and challenges in VLSI circuit reliability. IEEE Micro **4**, 14–19 (2003)
4. N.K. Jha, S. Gupta, *Testing of Digital Systems* (Cambridge University Press, 2003)
5. C. Constantinescu, Impact of Intermittent Faults on Nanocomputing Devices. In *DSN 2007 Workshop on Dependable and Secure Nanocomputing*, 2007
6. P.M. Wells, K. Chakraborty, G.S. Sohi, Adapting to intermittent faults in multicore systems. In *ACM Sigplan Notices*, *ACM*, vol. 43, 2008, pp. 255–264
7. M. Bushnell, V.D. Agrawal, *Essentials of electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits*, vol. 17. Springer Science & Business Media, 2000
8. R. Rajsuman, *Digital Hardware Testing: Transistor-Level Fault Modeling and Testing* (Artech House Inc, Norwood, MA, USA, 1992)
9. S. Gosh, T.J. Chakraborty, On behavior fault modeling for digital systems. J. Electron. Test. Theory Appl. **2**, 135–151 (1991)
10. R.J. Hayne. *Behavioral fault modeling in a VHDL synthesis environment*. Ph.D. Thesis, University of Virginia, 1999
11. L.-C. Wang, M.S. Abadir, Test generation based on high-level assertion specification for PowerPCTM microprocessor embedded arrays. J. Electron. Test. **13**(2), 121–135 (1998)
12. P.C. Ward, J.R. Armstrong, Behavioral Fault Simulation in VHDL. in *Proceedings of the 27th ACM/IEEE Design Automation Conference on* ACM, 1991, pp. 587–593
13. P. Banerjee, A model for simulating physical failures in MOS VLSI circuits. *Coordinated Science Laboratory Report no. CSG-13*, 1982
14. J.P. Hayes, Fault modeling. IEEE Des. Test Comput. **2**(2), 88–95 (1985)

15. T. Sridhar, J.P. Hayes, A functional approach to testing bit-sliced microprocessors. IEEE Trans. Comput. **100**(8), 563–571 (1981)
16. M.C. Hansen, J.P. Hayes, High-level Test Generation Using Physically-Induced Faults. in *Proceedings of VLSI Test Symposium on IEEE, 13th IEEE*, 1995, pp. 20–28
17. M. Michael, S. Tragoudas, ATPG tools for delay faults at the functional level. ACM Trans. Des. Autom. Electron. Syst. **7**(1), 33–57 (2002)
18. S.M. Thatte, J.A. Abraham, Test generation for microprocessors. IEEE Trans. Comput. **100** (6), 429–441 (1980)
19. D. Brahme, J.A. Abraham, Functional testing of microprocessors. IEEE Trans. Comput. **100** (6), 475–485 (1984)
20. M.-L. Li, P. Ramachandran, U.R. Karpuzcu, S.K.S. Hari, S.V. Adve, Accurate Microarchitecture-level Fault Modeling for Studying Hardware Faults. In *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, 2009, pp. 105–116
21. K. Christou, M.K. Michael, P. Bernardi, M. Grosso, E. Sánchez, M.S. Reorda. A Novel SBST Generation Technique for Path-delay Faults in Microprocessors Exploiting Gate-and RT-level Descriptions. in *26th IEEE VLSI Test Symposium (VTS 2008)*, 2008, pp. 389–394
22. A.J. Van de Goor, C.A. Verruijt, An overview of deterministic functional RAM chip testing. ACM Comput. Surv. **22**(1), 5–33 (1990)
23. R. Nair, S.M. Thatte, J.A. Abraham, Efficient algorithms for testing semiconductor random-access memories. IEEE Trans. Comput. **27**(6), 572–576 (1978)
24. A.J. Van de Goor, *Testing Semiconductor Memories: Theory and Practice* (Wiley, London, 1991)
25. John P. Hayes, Detection of pattern-sensitive faults in random-access memories. IEEE Trans. Comput. **100**(2), 150–157 (1975)
26. A. Krstic, K.-T. Cheng, *Delay fault testing for VLSI circuits*, vol. 14. Springer Science & Business Media, 1998
27. M. Sivaraman, A.J. Strojwas, *A Unified Approach for Timing Verification and Delay Fault Testing*. Springer Science & Business Media, 2012
28. S.N. Neophytou, M.K. Michael, S. Tragoudas, Functions for quality transition-fault tests and their applications in test-set enhancement. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **25**(12), 3026–3035 (2006)
29. G.L. Smith, Model for Delay Faults Based Upon Paths. in *ITC*, Citeseer, 1985, pp. 342–351
30. A. Krstic, K.-T. Cheng, S.T. Chakradhar, Primitive delay faults: identification, testing, and design for testability. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **18**(6), 669–684 (1999)
31. M. Sivaraman, A.J. Strojwas, Primitive path delay faults: identification and their use in timing analysis. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **19**(11), 1347–1362 (2000)
32. K. Christou, M.K. Michael, S. Neophytou, Identification of Critical Primitive Path Delay Faults Without any Path Enumeration. in *2010 28th VLSI Test Symposium (VTS) on IEEE*, 2010, pp. 9–14
33. T.C. May, M.H. Woods, A New Physical Mechanism for Soft Errors in Dynamic Memories. in *IEEE Reliability Physics Symposium, 1978. 16th Annual*, 1978, pp. 33–40
34. T.C. May, M.H. Woods, Alpha-particle-induced soft errors in dynamic memories. IEEE Trans. Electron Devices **26**(1), 2–9 (1979)
35. R.C. Baumann, Radiation-induced soft errors in advanced semiconductor technologies. IEEE Trans. Device Mater. Reliab. **5**(3), 305–316 (2005)
36. JEDEC Standard JESD89A, Measurement and reporting of alpha particle and terrestrial cosmic ray-induced soft errors in semiconductor devices. *JEDEC solid state technology association*, 2006
37. S. Mukherjee, *Architecture Design for Soft Errors* (Morgan Kaufmann, 2008)
38. M. Nicolaidis, *Soft errors in modern electronic systems*, vol. 41. Springer Science & Business Media, 2010

39. S.E. Diehl, A. Ochoa, P.V. Dressendorfer, R. Koga, W.A. Kolasinski, Error analysis and prevention of cosmic ion-induced soft errors in static CMOS RAMs. IEEE Trans. Nucl. Sci. **29**(6), 2032–2039 (1982)
40. P.E. Dodd, F.W. Sexton, Critical charge concepts for CMOS SRAMs. IEEE Trans. Nucl. Sci. **42**(6), 1764–1771 (1995)
41. R. Naseer, J. Draper, Parallel Double Error Correcting Code Design to Mitigate Multi-bit Upsets in SRAMs. in *34th European Solid-State Circuits Conference on IEEE, ESSCIRC 2008*, 2008, pp. 222–225
42. P.S. Ostler, M.P. Caffrey, D.S. Gibelyou, P.S. Graham, K.S. Morgan, B.H. Pratt, H.M. Quinn, M.J. Wirthlin, SRAM FPGA reliability analysis for harsh radiation environments. IEEE Trans. Nucl. Sci. **56**(6), 3519–3526 (2009)
43. T. Karnik, P. Hazucha, Characterization of soft errors caused by single event upsets in CMOS processes. IEEE Trans. Dependable Secure Comput. **1**(2), 128–143 (2004)
44. R.D. Schrimpf, D.M. Fleetwood, *Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices*, vol. 34. World Scientific, 2004
45. G. Georgakos, P. Huber, M. Ostermayr, E. Amirante, F. Ruckerbauer, Investigation of Increased Multi-bit Failure Rate Due to Neutron Induced SEU in Advanced Embedded SRAMs. in *2007 IEEE Symposium on VLSI Circuits*, 2007
46. M. Maniatakos, M. Michael, C. Tirumurti, Y. Makris, Revisiting vulnerability analysis in modern microprocessors. IEEE Trans. Comput. **64**(9), 2664–2674 (2015)
47. H. Belhaddad, R. Perez, M. Nicolaidis, R. Gaillard, M. Derbey, F. Benistant, Circuit Simulations of SEU and SET Disruptions by Means of an Empirical Model Built Thanks to a Set of 3d Mixed-mode Device Simulation Responses. in *Proceedings of RADECS*, 2006
48. H. Belhaddad, R. Perez, Apparatus and method for the determination of SEU and SET disruptions in a circuit caused by ionizing particle strikes, May 29 2007. US Patent App. 11/807,433
49. IROC Tech. TFIT Software. https://www.iroctech.com/solutions/transistorcell-level-fault-simulation-tools-and-services, 2016
50. A. Balasubramanian, B.L. Bhuva, J.D. Black, L.W. Massengill, RHBD techniques for mitigating effects of single-event hits using guard-gates. IEEE Trans Nucl Sci **52**(6), 2531–2535 (2005)
51. R.L. Shuler, A. Balasubramanian, B. Narasimham, B.L. Bhuva, P.M. O'Neill, C. Kouba, The effectiveness of tag or guard-gates in set suppression using delay and dual-rail configurations at 0.35 μm. IEEE Trans. Nucl. Sci. **53**(6), 3428–3431 (2006)
52. P.E. Dodd, M.R. Shaneyfelt, J.R. Schwank, G.L. Hash, Neutron-induced Latchup in SRAMs at Ground Level. in *Reliability Physics Symposium Proceedings, 2003. 41st Annual. 2003 IEEE International*, *IEEE*, 2003, pp. 51–55
53. Altera Corp. Altera FPGA Overview. https://www.altera.com/products/fpga/overview.html, 2016
54. Xilinx Inc, Xilinx FPGA Devices. http://www.xilinx.com/products/silicon-devices/fpga.html , 2016
55. M.B. Tahoori, E.J. McCluskey, M. Renovell, P. Faure. A Multi-configuration Strategy for an Application Dependent testing of FPGAs. in *Proceedings of 22nd IEEE VLSI Test Symposium*, 2004, pp. 154–159
56. M. Abramovici, C. Stroud, BIST-based Detection and Diagnosis of Multiple Faults in FPGAs. in *Proceedings of International Test Conference, 2000*, pp. 785–794, 2000
57. M. Renovell, J.M. Portal, J. Figueras, Y. Zorian. SRAM-based FPGA's: Testing the LUT/RAM Modules. in *Proceedings of International Test Conference*, 1998, pp. 1102–1111
58. M. Renovell, J.M. Portal, J. Figueras, Y. Zorian, Testing the interconnect of RAM-based FPGAs. IEEE Des. Test Comput. **15**(1), 45–50 (1998)
59. C. Stroud, S. Wijesuriya, C. Hamilton, M. Abramovici, Built-in self-test of fpga interconnect. in *Proceedings of International Test Conference,* 1998, pp. 404–411
60. X. Sun, J. Xu, B. Chan, P. Trouborst, Novel Technique for Built-in Self-test of FPGA Interconnects. in *Proceedings of International Test Conference,* 2000, pp. 795–803

61. M.B. Tahoori, S. Mitra, Application-independent testing of fpga interconnects. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **24**(11), 1774–1783 (2005)
62. L. Bauer, C. Braun, M.E. Imhof, M.A. Kochte, E. Schneider, H. Zhang, J. Henkel, H. J. Wunderlich, Test strategies for reliable runtime reconfigurable architectures. IEEE Trans. Comput. **62**(8), 1494–1507 (2013)
63. A. Cilardo, New techniques and tools for application-dependent testing of FPGA-based components. IEEE Trans. Industr. Inf. **11**(1), 94–103 (2015)
64. T.N. Kumar, F. Lombardi, A novel heuristic method for application-dependent testing of a SRAM-based FPGA interconnect. IEEE Trans. Comput. **62**(1), 163–172 (2013)
65. M. Tahoori, Application-dependent testing of FPGAs. IEEE Trans. Very Large Scale Integr. VLSI Syst. **14**(9), 1024–1033 (2006)
66. M.B. Tahoori, S. Mitra, Application-dependent delay testing of FPGAs. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **26**(3), 553–563 (2007)
67. M. Rebaudengo, M.S. Reorda, M. Violante, A New Functional Fault Model for FPGA Application-oriented Testing. in *Proceedings of 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2002. DFT 2002*, 2002, pp. 372–380
68. E. Chmelaf, FPGA Interconnect Delay Fault Testing. in *Proceedings of International Test Conference (ITC)*, 2003, vol 1, pp. 1239–1247
69. P.R. Menon, W. Xu, R. Tessier, Design-specific path delay testing in lookup-table-based FPGAs. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **25**(5), 867–877 (2006)
70. M. Psarakis, D. Gizopoulos, A. Paschalis, Test generation and fault simulation for cell fault model using stuck-at fault model based test tools. J. Electron. Test. **13**(3), 315–319 (1998)
71. A.J. Van De Goor, Using march tests to test SRAMs. IEEE Des. Test Comput. **10**(1), 8–14 (1993)
72. L.-T. Wang, C.-W. Wu, X. Wen, *VLSI Test Principles and Architectures: Design for Testability* (Academic Press, 2006)
73. W.K. Huang, F.J. Meyer, X.-T. Chen, F. Lombardi, Testing configurable LUT-based FPGA's. IEEE Trans. Very Large Scale Integr. Syst. **6**(2), 276–283 (1998)
74. S. Jamuna, V.K. Agrawal, Implementation of Bistcontroller for Fault Detection in CLB of FPGA. in *2012 International Conference on Devices, Circuits and Systems (ICDCS)*, 2012, pp. 99–104
75. S.-J. Wang, T.-M. Tsai, Test and Diagnosis of Faulty Logic Blocks in FPGAs. in *IEEE/ACM International Conference on Computer-Aided Design, 1997. Digest of Technical Papers*, 1997, pp. 722–727
76. M.B. Tahoori, Application-Dependent Testing of FPGA Interconnects. in *Proceedings of 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2003, pp. 409–416
77. H. Asadi, M.B. Tahoori, Analytical techniques for soft error rate modeling and mitigation of FPGA-based designs. IEEE Trans. Very Large Scale Integr. VLSI Syst. **15**(12), 1320–1331 (2007)
78. C. Bernardeschi, L. Cassano, A. Domenici, L. Sterpone, Assess: A simulator of soft errors in the configuration memory of SRAM-based FPGAs. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **33**(9), 1342–1355 (2014)
79. C. Bolchini, A. Miele, C. Sandionigi, Increasing Autonomous Fault-tolerant FPGA-based Systems' Lifetime. in *2012 17th IEEE European Test Symposium (ETS)*, 2012, pp.1–6
80. K. Morgan, M. Caffrey, P. Graham, E. Johnson, B. Pratt, M. Wirthlin, Seu-induced persistent error propagation in FPGAs. IEEE Trans. Nucl. Sci. **52**(6), 2438–2445 (2005)
81. E.S.S. Reddy, V. Chandrasekhar, M. Sashikanth, V. Kamakoti, N. Vijaykrishnan, Detecting SEU-caused routing errors in SRAM-based FPGAs. in *18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*, 2005, pp. 736–741
82. M. Violante, L. Sterpone, M. Ceschia, D. Bortolato, P. Bernardi, M.S. Reorda, A. Paccagnella, Simulation-based analysis of seu effects in SRAM-based FPGAs. IEEE Trans. Nucl. Sci. **51**(6), 3354–3359 (2004)

83. M. Ebrahimi, P.M.B. Rao, R. Seyyedi, M.B. Tahoori, Low-cost multiple bit upset correction in SRAM-based fpga configuration frames. IEEE Trans. Very Large Scale Integr. VLSI Syst. **24**(3), 932–943 (2016)
84. M. Lanuzza, P. Zicari, F. Frustaci, S. Perri, and P. Corsonello, A Self-hosting Configuration Management System to Mitigate the Impact of Radiation-induced Multi-bit Upsets in SRAM-based FPGAs. in *2010 IEEE International Symposium on Industrial Electronics*, 2010, pp. 1989–1994
85. P.M. B. Rao, M. Ebrahimi, R. Seyyedi, M.B. Tahoori, Protecting SRAM-based FPGAs Against Multiple Bit Upsets Using Erasure Codes. in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2014, pp. 1–6
86. A. Sari, M. Psarakis, Scrubbing-based SEU mitigation approach for systems-on-programmable-chips. in *2011 International Conference on Field-Programmable Technology (FPT)*, 2011, pp. 1–8
87. A. Sari, M. Psarakis, and D. Gizopoulos. Combining checkpointing and scrubbing in fpga-based real-time systems. in *VLSI Test Symposium (VTS), 2013 IEEE 31st*, pages 1–6, April 2013
88. C. Bolchini, A. Miele, M.D, Santambrogio, TMR and Partial Dynamic Reconfiguration to Mitigate SEU Faults in FPGAs. in *22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007)*, 2007, pp. 87–95
89. F.G. de Lima Kastensmidt, G. Neuberger, R.F. Hentschke, L. Carro, R. Reis, Designing fault-tolerant techniques for SRAM-based FPGAs. IEEE Des. Test Comput. **21**(6), 552–562 (2004)
90. J.M. Johnson, M.J. Wirthlin, Voter Insertion Algorithms for FPGA Designs Using Triple Modular Redundancy. in *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA'10, New York, NY, USA, 2010. ACM, pp. 249–258
91. F.L. Kastensmidt, L. Carro, *Fault-tolerance techniques for SRAM-based FPGAs*, vol. 1. Springer, Berlin
92. F. Lahrach, A. Doumar, E. Châtelet, A. Abdaoui, Master-slave TMR Inspired Technique for Fault Tolerance of SRAM-based FPGA. in *2010 IEEE Computer Society Annual Symposium on VLSI*, 2010, pp. 58–62
93. H.R. Zarandi, S.G. Miremadi, C. Argyrides, D.K. Pradhan, CLB-based Detection and Correction of Bit-flip Faults in SRAM-based FPGAs. in *2007 IEEE International Symposium on Circuits and Systems*, 2007, pp. 3696–3699
94. C. Bolchini, A. Miele, C. Sandionigi, A novel design methodology for implementing reliability-aware systems on SRAM-based FPGAs. IEEE Trans. Comput. **60**(12), 1744–1758 (2011)
95. C. Bolchini, C. Sandionigi, Fault classification for SRAM-based FPGAs in the space environment for fault mitigation. IEEE Embed. Syst. Lett. **2**(4), 107–110 (2010)
96. B.S. Gill, C. Papachristou, F.G. Wolff, A New Asymmetric SRAM Cell to Reduce Soft Errors and Leakage Power in FPGA. in *2007 Design, Automation Test in Europe Conference Exhibition*, 2007, pp. 1–6
97. S. Mitra, N. Seifert, M. Zhang, Q. Shi, K.S. Kim, Robust system design with built-in soft-error resilience. Computer **38**(2), 43–52 (2005)
98. M. Psarakis, A. Vavousis, C. Bolchini, A. Miele, Design and Implementation of a Self-healing Processor on SRAM-based FPGAs. in *2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2014, pp. 165–170
99. M. Sonza Reorda, M. Violante, C. Meinhardt, R. Reis, A Low-cost See Mitigation Solution for Soft-processors Embedded in Systems on Programmable Chips. in *2009 Design, Automation Test in Europe Conference Exhibition*, 2009, pp. 352–357
100. L. Benini, G. De Micheli, Networks on chips: a new soc paradigm. Computer **35**(1), 70–78 (2002)
101. É. Cota, A. de Morais Amory, M. Soares Lubaszewski, *Reliability, Availability and Serviceability of Networks-on-chip*. Springer Science & Business Media, 2011

102. G. De Micheli, L. Benini, *Networks on Chips: Technology and Tools* (Academic Press, 2006)
103. C. Nicopoulos, S. Srinivasan, A. Yanamandra, D. Park, V. Narayanan, C.R. Das, M.J. Irwin, On the effects of process variation in network-on-chip architectures. IEEE Trans. Dependable Secure Comput. **7**(3), 240–254 (2010)
104. Y. Zorian, Guest editor's introduction: what is infrastructure ip? IEEE Des. Test Comput. **19** (3), 3–5 (2002)
105. P.S. Bhojwani, R.N. Mahapatra, Robust concurrent online testing of network-on-chip-based socs. IEEE Trans. Very Large Scale Integr. VLSI Syst. **16**(9), 1199–1209 (2008)
106. A. Dalirsani, M.E. Imhof, H.J. Wunderlich, Structural software-based self-test of network-on-chip. in *2014 IEEE 32nd VLSI Test Symposium (VTS)*, 2014, pp. 1–6
107. C. Liu, K. Chakrabarty, Identification of error-capturing scan cells in scan-BIST with applications to system-on-chip. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **23**(10), 1447–1459 (2004)
108. V. lyengar, K. Chakrabarty, E.J. Marinissen, Test access mechanism optimization, test scheduling, and tester data volume reduction for system-on-chip. IEEE Trans. Comput. **52** (12), 1619–1632 (2003)
109. A. Manzone, P. Bernardi, M. Grosso, M. Rebaudengo, E. Sanchez, M.S. Reorda, Integrating BIST Techniques for On-line SoC Testing. in *11th IEEE International On-Line Testing Symposium*, 2005, pp. 235–240
110. J. Raik, V. Govind, R. Ubar, An External Test Approach for Network-on-a-chip Switches. In *2006 15th Asian Test Symposium*, 2006, pp. 437–442
111. B. Vermeulen, J. Dielissen, K. Goossens, C. Ciordas, Bringing communication networks on a chip: test and verification implications. IEEE Commun. Mag. **41**(9), 74–81 (2003)
112. K. Stewart, S. Tragoudas, Interconnect Testing for Networks on Chips. In *24th IEEE VLSI Test Symposium*, 2006, 6 pp
113. A.M. Amory, E. Briao, E. Cota, M. Lubaszewski, F.G. Moraes, A Scalable Test Strategy for Network-on-chip Routers. in *IEEE International Conference on Test, 2005*, 2005, pp.9–599
114. C. Grecu, P. Pande, Baosheng Wang, A. Ivanov, R. Saleh, Methodologies and Algorithms for Testing Switch-based NoC Interconnects. in *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*, 2005, pp. 238–246
115. J. Raik, R. Ubar, V. Govind, Test Configurations for Diagnosing Faulty Links in NoC Switches. in *12th IEEE European Test Symposium (ETS'07)*, 2007, pp. 29–34
116. A. Alaghi, N. Karimi, M. Sedghi, Z. Navabi, Online NoC Switch Fault Detection and Diagnosis Using a High Level Fault Model. in *IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007)*, 2007, pp. 21–29
117. D. Bertozzi, L. Benini, G. De Micheli, Error control schemes for on-chip communication links: the energy-reliability tradeoff. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **24**(6), 818–831 (2005)
118. S. Murali, T. Theocharides, N. Vijaykrishnan, M.J. Irwin, L. Benini, G. De Micheli, Analysis of error recovery schemes for networks on chips. IEEE Des. Test Comput. **22**(5), 434–442 (2005)
119. É. Cota, F.L. Kastensmidt, M. Cassel, M. Herve, P. Almeida, P. Meirelles, A. Amory, M. Lubaszewski, A High-fault-coverage Approach for the Test of Data, Control and Handshake Interconnects in Mesh Networks-on-chip. IEEE Trans. Comput. **57**(9), 1202–1215 (2008)
120. C. Grecu, A. Ivanov, R. Saleh, P.P. Pande, Testing network-on-chip communication fabrics. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **26**(12), 2201–2214 (2007)
121. E. Cota, F.L. Kastensmidt, A. Amory, M. Cassel, M. Lubasweski, P. Meirelles, Redefining and testing interconnect faults in mesh NoCs. in *IEEE International Test Conference*, 2007, pp. 1–10
122. C. Bolchini, M. Carminati, M. Gribaudo, A. Miele, A lightweight and open-source framework for the lifetime estimation of multicore systems. in *Proceedings of International Conference on Computer Design*, 2014, pp. 166–172

123. Joint Electron Device Engineering Council, Failure Mechanisms and Models for Silicon Semiconductor Devices. Technical Report JEP122G, 2011
124. Y. Xiang, T. Chantem, R.P. Dick, X.S. Hu, L. Shang, System-level Reliability Modeling for MPSoCs. in *Proceeding of Conferences on Hardware/Software Codesign and System Synthesis (CODES)*, 2010, pp. 297–306
125. L. Huang, Q. Xu, AgeSim: A Simulation Framework for Evaluating the Lifetime Reliability of Processor-based SoCs. in *Proceedings of Conference on Design, Automation Test in Europe (DATE)*, 2010, pp. 51–56
126. L. Huang, F. Yuan, Q. Xu, On Task allocation and scheduling for lifetime extension of platform-based MPSoC designs. IEEE Trans. Parallel Distrib. Syst. **22**(12), 2088–2099 (2011)
127. I. Ukhov, M. Bao, P. Eles, Z. Peng, Steady-state Dynamic Temperature Analysis and Reliability Optimization for Embedded Multiprocessor Systems. in *Proceedings of Design Automation Conference (DAC)*, 2012, pp. 197–204
128. JEDEC Solid State Technology Association. http://www.jedec.org
129. J.R. Black, Electromigation—a brief survey and some recent results. IEEE Trans. Electron Devices **16**(4), 338–347 (1969)
130. J. Srinivasan, S.V. Adve, P. Bose, J.A. Rivers, Lifetime reliability: toward an architectural solution. IEEE Micro **25**(3), 70–80 (2005)
131. Y. Zhang, M.L. Dunn, K. Gall, J.W. Elam, S.M. George, Suppression of inelastic deformation of nanocoated thin film microstructures. AIP J. Appl. Phys. **95**(12), 8216–8225 (2004)
132. M. Ciappa, F. Carbognani, W. Fichtner, Lifetime prediction and design of reliability tests for high-power devices in automotive applications. IEEE Trans. Device Mater. Reliab. **3**(4), 191–196 (2003)
133. B.L. Amstadter, *Reliability Mathematics: Fundamentals, Practices, Procedures* (McGraw-Hill, NY, 1977)
134. Joint Electron Device Engineering Council, Method for Developing Acceleration Models for Electronic Component Failure Mechanisms. Technical Report JESD91A, 2003
135. Y. Zhang, M.L. Dunn, K. Gall, J.W. Elam, S.M. George, The electromigration failure distribution: the fine-line case. J. Appl. Phys. **69**(4), 2117–2127 (1991)
136. R. Degraeve, G. Groeseneken, R. Bellens, M. Depas, H.E. Maes. A Consistent Model for the Thickness Dependence of Intrinsic Breakdown in Ultra-thin OXIDEs. in *International Electron Devices Meeting*, 1995, pp. 863–866
137. H. Liu, Reliability of a load-sharing k-out-of-n: G system: non-iid components with arbitrary distributions. Trans. Reliab. **47**(3), 279–284 (1998)
138. S.D. Dowling, D.F. Socie, Simple rainflow counting algorithms. Int. Journal of Fatigue **4**(1), 31–40 (1983)
139. N.E. Dowling. *Mechanical Behavior of Materials*. (Pearson/Prentice Hall, 3rd ed., 2007)
140. K.S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications* (Wiley, Chichester, UK, 2nd ed., 2002)
141. S. Corbetta, D. Zoni, and W. Fornaciari. A Temperature and Reliability Oriented Simulation Framework for Multi-core Architectures. in *International Symposium on VLSI,* 2012, pp. 51–56
142. E. Karl, D. Blaauw, D. Sylvester, T. Mudge, Multi-mechanism reliability modeling and management in dynamic systems. Trans. VLSI Syst. **16**(4), 476–487 (2008)
143. C. Bolchini, L. Cassano, A. Miele, Lifetime-aware Load Distribution Policies in Multi-core Systems: An In-depth Analysis. in *Proceedings of International Conference on Design, Automation and Testing in Europe (DATE)*, 2016, pp. 804–809
144. Reliability-metric varieties and their relationships. in *Proceedings of Reliability and Maintainability symposium*, 2001
145. K.S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications* (Wiley, 2nd ed., 2002)

Application Scenarios

Hans Manhaeve and Viacheslav Izosimov

Abstract To illustrate the manufacturing threats addressed in Chap. "Manufacturing Threats" and the dependability threats elaborated in Chap. "Dependability Threats", this chapter will address a number of application cases from different domains, such as automotive, railroad and transportation, air and space and medical, where safety-critical and reliable operations are key. It will address current practices deployed in these different domains and highlights the risks involved when the effects of the ever-scaling technologies and related design techniques on system reliability are not properly taken into consideration. Finally, the chapter will discuss hardware security, which is a common challenge in all the domains.

1 Dependability and Advancement of Hardware

Applications pose various requirements on hardware in terms of reliability and performance. Often, these requirements are inter-related. It, for example, is believed that advanced electronics are not reliable enough to be used in aerospace. Due to long lifecycle requirements some installations in the energy domain can be stuck to 30–40-year-old electronics. The machine domain often utilizes hardware solutions already available on the market, for example, from the surface transportation domain, by picking up those that suit particular needs and are proven to be robust. The surface transportation domain, due to a huge mass market (automotive in particular) and great attractiveness for hardware manufacturers is able to impose own strict dependability requirements. Often, automotive manufacturers utilize the most successful solutions from the consumer electronics domain, making them

H. Manhaeve (✉)
Ridgetop Europe, Bruges, Belgium
e-mail: hans.manhaeve@ridgetop.eu

V. Izosimov
KTH Royal Institute of Technology, Stockholm, Sweden

V. Izosimov
Semcon Sweden AB, Stockholm, Sweden

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_3

hardened and more robust. The Medical and healthcare domain, on the other hand, is in a very different position. Saving lives is a great driver toward hardware implementations. As an example: saving 99 lives out of 100 even if 1 life is missed due to hardware dependability issues will always pay off. It is not possible to wait a few more years until the hardware becomes more mature and miss all the 100 lives at stake. Finally, consumer electronics is an excellent lab where new hardware solutions emerge, where performance is a key and the mass market is enormous. Requirements on dependability in consumer electronics are not as critical as in other domains, with automotive as an example. Still consumers would often prefer to have features that function most of the time. In some cases, hardware manufactures, especially those that are also present in the transportation domain, introduce dependability solutions into their consumer electronics hardware. By that, they are able to test the dependability solutions that will be eventually needed and increase customer satisfaction without taking too much of a risk. From our point of view, consumer electronics is a great enabler of the technological advances that will be eventually used by all the domains. For example, the 80186 processers that once upon a time emerged in consumer electronics, are still used in the aerospace domain for designing control systems. Another emerging enabler of hardware advancement is the surface transportation domain; in particular, automotive that is in need of robust yet inexpensive hardware and ready to invest into development. Hardware manufacturers should target fist of all these two markets in order to even succeed in other domains. Figure 1 summarizes the placement of the applications domains in a two-dimensional chart. The Y-axis scales dependability requirements and the X-axis scales advancement in hardware.

We also would like to highlight an interesting new trend in hardware development: the large number of Open Source Hardware solutions emerging on the market. These solutions target a new emerging consumer electronics market in the Internet of Things and support prototyping of Cyber-Physical Systems, with lots of start-ups trying to propose new products. The Open Source Hardware allows to quickly

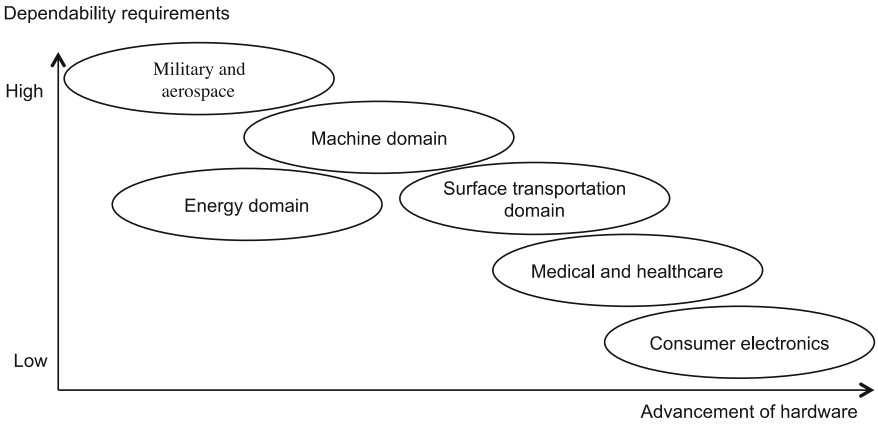


Fig. 1 Dependability and advancement of hardware in different application domains

assemble a product for demonstration to prospective customers and producing small prototype series. Although many of these board are of a purely consumer electronics type, there are a few that already have dependability-related mechanisms, for example, lockstep, memory protection, protected internal flashes, etc. We see this new emerging market also as a very interesting target for hardware manufacturers since it enables flexibility and a greater freedom of solutions. Eventually, these start-ups and hobby projects will come out with novel products, potentially, game changing, and the underlying hardware platforms will be winning on the market.

Another important aspect with respect to system deployment and reliability assurance is the impact of the actual usage profile versus the intended usage profile on the fitness and the lifetime of the product [1]. Some devices may significantly deviate from the "average profile" and it is a great challenge for a hardware manufacturer to ensure that even "corner" cases are appropriate taken care of. If these corner cases are important for the overall business, special-purpose hardened versions of devices may be manufactured, thus, tailoring customers' needs best.

2 Military and Aerospace

Military and aerospace applications are among the ones with the most stringent reliability requirements. Typically, the reliability estimates for these systems rely on standards such as MIL-STD-217, MIL-STD-883, Telcordia, and others, which are statistical measures that consider macro-level part types and general populations of components on the boards or subsystems. A key problem is that the statistical nature of this approach does not adequately address the harsh use environment that is actually experienced by the electronic-based equipment, such as latent damage from component value drift from age, radiation exposure, and temperature extremes. To overcome this, recent applications have involved condition-based monitoring of the individual Mil/Aero subsystems. This process consists of sensing, data conditioning, and the application of state estimators to yield State-of-Health (SoH) and Remaining Useful Life (RUL). Failure of electronic components serving equipment for military and aerospace deployment may potentially have devastating consequences for the mission. A particular issue, unique to these applications, is availability and more important obsolescence of the components used to build the system. Due to long system, board and component lifetime requirements and the tremendous certification efforts required to qualify systems and ensure system reliability and safety, it may even happen that components become obsolete even during a design process (which can be of 20–30 years in some cases) or during the operation of the system (which can be of 30–40 years in many cases). Addressing obsolescence and implementing strategies for replacement of obsolete components as part of the design cycle is hence necessary. Simply replacing components with others or modifying the design to accommodate new components is mostly not possible due to prohibiting re-certification effort involved when doing so. To address such issues, a large number of initiatives are presently ongoing [2]; for example, the introduction of so called D-E-R (Die Extraction and Repackaging) components helps to address such issues at minimum effort required [3–8]. This involves extracting die from (new) packages that are not fit to the design, having them repackaged in the desired package so that they are looking exactly like the ones that are featured in the design, and having them retested and qualified. Another solution to the problem is also reserving a large quantity of components and storing them for a longer period of time. Vendors will also solicit such "Last Time Buy" opportunities to their customers. Apart from the cost issues the storage of semiconductor components over longer periods of time, may, however, impact quality of the raw material and deviate properties of the components. In general, in this domain, verification, qualification, and device reliability are of ultimate importance. This imposes process requirements on quality assurance in development and manufacturing, even including packaging of the components. In particular use of complex components, with multilayered and multi-stack structures, are particularly challenging for qualification in these domains [9]. Even multi-core hardware is in contrast to application in other domains still not yet fully adapted in the military and aerospace fields. On the other hand, fault tolerance solutions in hardware such as hardening are well developed and accepted within the defense and aerospace domain. Good examples of microprocessors initially designed for aerospace applications are the LEON family, with the LEON III and LEON IV as prime ones [10, 11]. Self-repair solutions are common for satellites that have to deal with aging of electronic components when extensively exposed to the impacts of harsh space environment, including huge ambient temperature variations, impacts of a variety of charged particles from deep space and the Sun bombarding the electronics with particle rains and solar flares.

3 Surface Transportation Domain

This section focusses on the reliability and dependability aspects and requirements for electronics targetted for use in surface transportation domain applications, from automotive to marine and railroad, whereas in the case of military and avionics applications redundancy plays a substantial role, in the transportation domain the focus of the dependability work is often directed towards monitoring systems and taking actions upon evidence of need [12, 13]. Health monitoring and health status assessment, in general, is studied for different case studies, from drive-by-wire [14] to sensor monitoring [15] and observation systems [16]. For example, experiments have been conducted using wireless rotational and vibrational sensors to investigate the reliability and health status of rolling material as well as the tracks they ride on [17, 18]. For this purpose, rotational and vibrational sensors were attached to the axels and wheels of the test vehicles. Analysis of the sensor data showed that such approaches allow to monitor the health status of both the rolling material as well as the tracks they ride on.

In the transportation domain, it is often about choosing the right main function and equipment, with the following instantiation of the observation tool [ISO 26262-3, EN 50126]. In case of failure, degraded mode or transitions into safe states will be activated. We will further study monitoring for marine systems, railroads, automotive vehicle, and commercial trucks and buses. These applications similar to military and avionics applications have specific qualification requirements. Qualification, however, permits decomposition of functionality from more critical to less critical and a high emphasis is placed on monitoring [ISO 26262-9]. One particular aspect, often accounted for in certifications, is mass production. For some applications, such as railroad and electrical vehicles, high voltages are used which require particular considerations [19]. Lifetime expectancy can be in some cases as long as that required in military and avionics, in particular, for trains and marine systems, typical lifetime expectancies are 30–40 years. In automotive systems, passenger vehicles usually have a shorter required life expectancy of 15 years, while commercial vehicles may have requirements on lasting for up to 30 years. In the transportation domains, systems are often produced continuously and are constantly upgraded, which is quite different from military and avionics. Mass-market requirements, significant in particular for passenger vehicles, have substantial implications on the calculation of reliability since probabilities of failures are multiplied with the number of vehicles.

4 Energy Domain

This section addresses the reliability and dependability aspects and requirements for applications in the energy domain. These applications are often characterized by continuous evolutions, e.g., the system evolves over a longer period of time in continuous modifications [20–24]. Many of the systems in the energy domain are in constant need for maintenance and the reduction of the need for maintenance is considered as exceptionally advantageous [25]. For example, electronics for deep sub-sea applications in the oil and gas industry are subject to expensive yearly maintenance checks and replacements. Being able to reduce maintenance actions whilst being able to guarantee reliable system operation is of high interest, which poses requirements on both reliability and packaging [26].

Without regular maintenance, systems will experience constant degradation of performance with the potential of sudden breakdown. For example, solar panels can be broken after years, with their efficiency degraded over time, thus requiring replacements [27]. There is a wide dispersion in safety and reliability requirements in the energy domain from the simple solar panels that are part of a household to the control of nuclear power plants. Some of the applications in the energy domain are exposed to the same requirements on hardening of electronic components as in the military and avionics domain. For example, in case of nuclear power plants, safety requirements are high, resulting in the need of multiple redundancies [28, 29]. Many of the applications in the energy domain require certifications and regularly re-certifications of the installations. Extension of the lifetime and extensive maintenance procedures are common in these applications. One particular example of deteriorating components is the one of batteries used for energy storage [30]. Their usage may lead to potential problems if the maintenance and reliability of the constructions and the electronic components used are not thoroughly considered. One well-known example of batteries leading to problems is the Dreamliner example [31]. If care is taken, batteries can be made robust and reusable across multiple domains. For example, they can be used in storing energy of electric bikes, automotive vehicles, in houses, etc. Dealing with energy storages and other devices in the energy domain often involves considering threats and risks. As an example, roof solar panels are often a threat for firefighters. For large systems, such as power grids, a great amount of threats and risks have to be accounted for, both in operating these systems and as well as in considering consequences that are possible if these systems break.

5 Medical and Healthcare

This section addresses the reliability and dependability aspects and requirements for medical and healthcare related application. Distinction is made between electronics for applications intrusive to the human body and for nonintrusive applications, mostly for hospital care and used in diagnosis and patient treatment. In case of implants, both safety and reliable operation are of high importance. Life lasting is a requirement as well as that these devices have to survive during the whole target's lifetime. Pacemakers are becoming more and more common implants, with a number of new devices coming up to the market, for example, neurostimulators (attached into the brain to treat diseases such as epilepsy and brain tumor). Besides the topic of health supporting devices, observation-type devices are also important to consider. Non-reliable devices or malfunctioning of observation devices might lead to wrong conclusions and resulting wrong actions, even if they do not affect the human body directly. The consequences of their malfunction can also be harmful in both short and longer runs. For example, common devices these days are those used for facilitating shaping of the human body. Despite not being even classified as a health application, they may have some issues with respect to their malfunctions, such as, causing painful muscle contraction in case of electrical malfunction, although not being harmful in general. Further, we consider monitoring devices and intrusive devices in hospital treatment that have a number of dependability requirements, both during operation—fail-operational and in case of diagnosis—safety requirements on the operational range—for example, the X-ray machine should not overshoot the radiation dose. Finally, Medical (Device) Cyber-Physical Systems is a new direction toward interactive infrastructure-interconnected medical systems [32]. Collaboration between medical devices is becoming more and more common these days with many of the medical devices interconnected with each other, and interacting with environments, doctors, nurses and patients. The MD-CPSs have particular concerns similar to what of larger systems such as power grids, with respect to their maintenance and potential consequences of malfunction. The Medical and Healthcare aspects will be discussed further in Chap. "Application Specific Solutions".

6 Machine Domain

The reliability and dependability aspects and requirements for electronics serving the Machine domain is another application area of interest to bring up for an application case. Machines are tools for workers, and not intended for use by the general public. It is often assumed that workers have been receiving a proper training for operation of these machines but reality may be different. Workers working every day with a machine are exposed to potential failures in this machine and resulting hazards, which increases dependability and, in particular, safety requirements. Some machines are stationary with static installations, e.g., on factories, which can be seen sometimes similar to the energy domain systems. Some machines are mobile, with operation in some cases even on public roads. In the latter case, similarities can be found with transportation domain. Many of the machines have certification and qualification requirements, in some cases very harsh, in some cases less strict [33, 34]. A particular consideration should be given to hand-held machines [35]. Their safety is a particular challenge since their operation directly depends on the way the operator performs its task. Hence, safety requirements and certifications on these machines can be harsh in many cases. In the machine domain, different solutions in dependability from different domains, for example, transportation or energy domain, while at the same time they are often subjects to certification. Many machines involve cyber-physical properties having a distributed nature of operation, are exposed to the environment and operate with the human operators directly or indirectly involved [36–39].

7 Consumer Electronics

The last application domain for which reliability and dependability aspects and requirements for electronics in this chapter are considered is the domain of the consumer electronics. For consumer electronics, there is a growing awareness in reliability, and safety, and that machines should permit some user operations while they should not permit others. The technology-related aspect is of a great interest in the consumer electronics since this domain is often in forefront of the technological development and the playfield for the introduction of new technologies. In this domain, awareness is growing against impacting functionality and availability of electronic components [40]. We will consider an example from a large electronic supplier that started recently investigations on this subject. Being on technology forefront, devices are becoming smaller and smaller, more and more sensitive to handling issues, for example, when are put into packages, in general, becoming more sensitive for physical stress, cracks, and alike [41–45]. Hence, it is becoming important to design solutions for die cracking. For example, a physical defect may "eat up" a wire over time and lead to failures that are not covered at time zero investigations (when detection is usually possible) but later after being in operation for some time. Reliability issues related to technology often show up early in this domain, and manufacturers attempt to provide solutions to increase reliability not because of safety issues in the application but to provide a sufficient level of performance and maintain brand status. Sometimes, even some safety-critical applications are considered in consumer electronics, with automotive passenger vehicles as one example (belonging to the transportation domain at the same time). Many of the other domains, discussed above, utilize findings and experiences of dealing with the new technologies from the consumer domain. Automotive is one example bordering to the transportation domain. Multimedia solutions coming into the avionics may facilitate introducing of new technologies into this very conservative field, as a result of customer demands on new functionality. This can be used systematically, in fact, in many application fields, for early testing of even domain-specific dependability solutions [46, 47]. These solutions can be introduced into mass-market electronics for increase of the performance; by failing, nothing would be harmed and at same time allowing gathering statistical information. We consider consumer electronics as a playground for new dependability solutions, and as early adopters of new technology that would then be introduced to other more conservative domains.

8 Protection Against Counterfeiting and Different Hardware Security Issues

Finally, in addition to discussion of dependability issues related to reliability and safety, security aspects should not be underestimated. Security is becoming increasingly important, in principle, in all domains. Security is also special due to the fact that security attacks and methods tend to spread quickly between multiple domains [48–54]. With all the electronic systems, in all the domains, becoming extremely interconnected, the same security requirements should apply across domains despite of the particular hardware solutions used. Even old hardware in the energy domain or in an airplane must be able to withstand modern and sophisticated security attacks. In this section, focus will be on particular types of attack vectors, namely attacks initiated with or by the hardware. One example of such attacks is the use of Hardware Trojans that can be embedded somewhere in the manufacturing chain, and then activated when, for example, the hardware is placed into an airplane.

Counterfeiting is a serious problem impacting customers and producers in the global economy. There are two separate issues: the economic impacts from purchasing a counterfeit consumer retail product, and the health risks associated with a consumer purchasing and consuming a counterfeit pharmaceutical product. This effort focuses on the consumer retail products only, as the detection methods for the two problem spaces are very different [55–63].

Among the most serious and urgent issues for the defense and intelligence communities are the presence in the supply chain of counterfeit electronic components, some of which are used in mission-critical applications. Despite numerous concepts and strategies to reliably detect these components that have been pursued, a proven, practical and streamlined solution to the problem still remains elusive.

Product counterfeiting is a form of consumer fraud: a counterfeit product is sold, pretending to be something that it is not. As a result, most product counterfeiting is considered to be criminal in nature under typical trade conventions. The key technical challenge to be considered is how to differentiate a possibly visibly identical counterfeit product from the authentic item. One of the key observables lending itself to automated-machine detection is testing for use of inferior-grade materials.

In general, methodologies exclusively based on electrical characterization lack the required throughput, while those mainly relying on visual screens typically fail to provide the required level of certainty. Moreover, other technologies that have involved embedding features within the component to prove its authenticity when interrogated tend to reduce component performance.

Various research programs are running seeking for appropriate solutions to address these issues and minimize the negative impact of counterfeiting [64–71].

References

1. http://www.ridgetopgroup.com/products/advanced-diagnostics-and-prognostics/
2. https://www.sbir.gov
3. AF161-142: https://www.sbir.gov/sbirsearch/detail/870421—Integrated Circuit (IC) Die Extraction and Reassembly
4. Electronic Circuits-Preserving Technique for Decapsulating Plastic Packages. 1987. IBM Technical Disclosure Bulletin **30**(6), 446–447 (1987)
5. United States Patent 6,429,028, 6 August 2002 DPA Labs, Incorporated (Simi Valley, CA)
6. Patented DPEM Process for Die Removal DPA Components International (Simi Valley, CA) http://www.dpaci.com/patented-dpem-process-for-die-removal.html
7. Global Circuit Innovations Website: http://www.gci-global.com/Global Circuit Innovations (Colorado Springs, CO)
8. Die extraction strategy solves DMSMS challenges global circuit innovations (Colorado Springs, CO) http://www.cotsjournalonline.com/articles/view/102446
9. http://grouper.ieee.org/groups/3Dtest/
10. https://en.wikipedia.org/wiki/LEON
11. http://www.gaisler.com/index.php/products/processors
12. X. Chen et al, Application of software watchdog as a dependability software service for automotive safety relevant systems, DSN (2007)
13. K. Tindell, H. Kopetz, F. Wolf, R. Ernst, Safe automotive software development, DATE (2003)
14. R. Isermann, R. Schwarz, S. Stolzl, Fault-tolerant drive-by-wire systems. IEEE Control Sys. **22**(5), (2002)

15. D. Capriglione, C. Liguori, A. Pietrosanto, Analytical redundancy for sensor fault isolation and accommodation in public transportation vehicles, IEEE Transactions on Instrumentation and Measurement **53**(4), (2004)
16. D. Reinhardt, G. Morgan, An embedded hypervisor for safety-relevant automotive E/E-systems, 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014). (2014)
17. http://www.ridgetopgroup.com/products/advanced-diagnostics-and-prognostics/ridgetop-sensors/rotosense/
18. http://www.ridgetopgroup.com/products/advanced-diagnostics-and-prognostics/sentinel-motion/railsafe-integrity-analysis-system/
19. D. Doan et al., Railroad tracks as a potential source of ignition in hazardous (Classified) Locations, 2007 IEEE Petroleum and Chemical Industry Technical Conference. (2007)
20. F. Blaabjerg, R. Teodorescu, M. Liserre, A.V. Timbus, Overview of control and grid synchronization for distributed power generation systems. IEEE Trans. Ind. Electron. **53**(5) (2006)
21. J.M. Carrasco et al., Power-Electronic Systems for the Grid Integration of Renewable Energy Sources: A Survey, IEEE Trans. Ind. Electron. **53**(4) (2006)
22. A. Ipakchi, F. Albuyeh, Grid of the future, IEEE Power Energy Mag. 7(2) (2009)
23. IEEE Guide to the Collection and Presentation of Electrical, Electronic, Sensing Component, and Mechanical Equipment Reliability Data for Nuclear-Power Generating Stations
24. R. Karki, P. Hu, R. Billinton, Reliability evaluation considering wind and hydro power coordination, IEEE Trans. Power Sys. **25**(2) 2010
25. J. Ribrant, L. Margareta Bertling, Survey of failures in wind power systems with focus on swedish wind power plants during 1997–2005. IEEE Trans. Energy Convers. **22**(1) (2007)
26. D.R.M. Woo et al., Extremely High Temperature and High Pressure (x-HTHP) Endurable SOI device & Sensor packaging for deep sea, oil and gas applications, IEEE Electronics Packaging Technology Conference (EPTC), 2014
27. Y. Wang, P. Zhang, W. Li, N.H. Kan'an, Comparative analysis of the reliability of grid-connected photovoltaic power systems, IEEE Power Energy Soc. Gen. Meet. (2012)
28. R. Dorr, F. Kratz, J. Ragot, F. Loisy, J.-L. Germain, Detection, isolation, and identification of sensor faults in nuclear power plants. IEEE Trans. Control Sys. Technol. **5**(1), 1997
29. K.B. Misra, Optimum reliability design of a system containing mixed redundancies. IEEE Trans. Power Appar. Sys. **94**(3) (1975)
30. S. Duryea, S. Islam, W. Lawrance, A battery management system for stand-alone photovoltaic energy systems. IEEE Ind. Appl. Mag. **7**(3) (2001)
31. Dreamliner: Boeing 787 planes grounded on safety fears. *BBC News*. January 17, 2013. Retrieved 21 Jun 2016
32. High-Confidence Medical Devices: Cyber-Physical Systems for 21st Century Health Care, A Research and Development Needs Report, High Confidence Software and Systems Coordinating Group, The Networking and Information Technology Research and Development (NITRD) Program, U.S. Federal Government, February 2009. Retrieved 26 Oct 2016
33. Directive 2006/42/EC
34. Occupational Safety and Health Standards, 1910.212
35. R. Morello, C. De Capua, A. Meduri, A Wireless Measurement System for Estimation of Human Exposure to Vibration During the Use of Handheld Percussion Machines, IEEE Trans. Instrum. Meas. **59**(10) (2010)
36. J.H. Graham, J.F. Meagher, S.J. Derby, A safety and collision avoidance system for industrial robots. IEEE Trans. Ind. Appl. **IA-22**(1) (1986)
37. J. Fryman, B. Matthias, Safety of industrial robots: from conventional to collaborative applications, robotics. 7th German Conference on ROBOTIK 2012
38. Volvo Construction Equipment, QUALITY, ENVIRONMENTAL CARE & SAFETY, volvoce.com, Retrieved 21 Jun 2016

39. Caterpillar Safety Services, Cultural Transformation. Operational Excellence, cat.com, Retrieved 21 Jun 2016
40. H. Manhaeve, A. Evans, M. Dubois, Can future system reliability problems be solved without deep understanding of technology?, Panel Member, Panel discussion at the 5th Workshop on Design for Reliability (DFR 2013), Berlin, Germany, 21–23 Jan 2013
41. H. Manhaeve, E. Mikkola, Semiconductor failure modes and mitigation for critical systems, ISQED 2012 Embedded Tutorial, The International Symposium on Quality Electronic Design (ISQED), Santa Clara, Ca, USA, 19–21 Mar 2012
42. http://www.ridgetopgroup.com/products/precision-instruments/prochek-for-wafer-level-reliability-wlr/
43. http://www.ridgetopgroup.com/products/advanced-diagnostics-and-prognostics/sentinel-interconnect/sj-bist-intermittency-detection/
44. http://www.ridgetopgroup.com/products/advanced-diagnostics-and-prognostics/sentinel-interconnect/tsv-bist/
45. H. Manhaeve, P. Davies, I.C. Yield, Reliability and prognostic methods using nanoscale test structures, DATE 2010 tutorial—part of the annual IEEE computer society TTTC Test Technology Educational Program (TTEP), Design automation and test in europe conference (DATE2010), Dresden, Germany, 8 Mar 2010
46. H. Manhaeve, Dynamic Supply Current Signature (Iddcs) Analysis, implementation and application benefits, in *Proceedings of the 11th European Manufacturing Test Conference (EMTC) and CAST Workshop*, Dresden, Germany, 8 Oct 2009
47. H. Manhaeve, Getting the best out of current testing, ZuE2010 Tutorial, Zuverlässigkeit und Entwurf, 4 GMM/GII/ITG-Fachtagung, Bildungszentrum Wildbach Kreuth, Germany, 13–15 Sept 2010
48. L. Batina, N. Mentens, B. Preneel, I. Verbauwhede, Balanced point operations for side-channel protection of elliptic curve cryptography. IEE Proc. Inf. Secur. **152**(1), 57–65 (2005)
49. J. Lano, N. Mentens, B. Preneel, I. Verbauwhede, Power analysis of synchronous stream ciphers with resynchronization mechanism. Int. J. Intell. Inf. Technol. Appl. **1**(2), 327–333 (2008)
50. A. Sadeghi, D. Naccache (eds.), *Towards hardware-intrinsic security, foundations and practice* (Springer, Heidelberg, 2010)
51. J. Vliegen, N. Mentens, I. Verbauwhede, A single-chip solution for the secure remote configuration of fpgas using bitstream compression. 2013 International Conference on Reconfigurable Computing and FPGAs, ReConFig 2013. RECONFIG 2013, IEEE Computer Society 2013, 09–11 Dec 2013
52. J. Vliegen, N. Mentens, I. Verbauwhede, Secure, remote, dynamic recon-figuration of FPGAs. ACM Trans. Reconfig. Technol. Sys. **7**(4), art.nr. 35 (2015)
53. J. Vliegen, D. Koch, N. Mentens, D. Schellekens, I. Verbauwhede, Practical feasibility evaluation and improvement of a pay-per-use licensing scheme for hardware IP cores in Xilinx FPGAs. J. Cryptogr. Eng. **5**(2), 113–122 (2015)
54. E.J. Marinissen, Y. Zorian, M. Konijnenburg, C.-T. Huang, P.-H. Hsieh, P. Cockburn, J. Delvaux, V. Rozic, B. Yang, D. Singelee, I. Verbauwhede, C. Mayor, R. van Rijsinge, C. Reyes, Special session: iot: source of test challenges, 21st IEEE Europe-an Test Symposium, Amsterdam, The Netherlands, 23–27 May 2016
55. U. Guin, K. Huang, D. DiMase, J. Carulli, Counterfeit integrated circuits: a rising threat in the global semiconductor supply chain. 2014 in *Proceedings of the IEEE* **102**(8)
56. J.M. Bryan, I.R. Cohen, O. Guzelsu, Inquiry into counterfeit electronic parts in the department of defense supply chain, Report of the Committee on Armed Services, United States Senate (2012). http://www.armed-services.senate.gov/download/inquiry-into-counterfeit-electronic-parts-in-the-department-of-defense-supply-chain
57. K. Huang, J.M. Carulli, Y. Makris, Parametric counterfeit ic detection via support vector machines. in *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, pp. 7–12, Oct 2012

58. V. Pathak, Improving supply chain robustness and preventing counterfeiting through authenticated product labels. in *Proceedings of IEEE International Conference on Technologies for Homeland Security*, pp. 35–41, Nov 2010
59. J. Federico, Detecting counterfeit electronic components. Evaluation engineering: instrumentation test report, 2009. http://www.njmetmtl.com/EE%2009_09_w.pdf
60. M. Tehranipoor, H. Salmani, X. Zhang, X. Wang, R. Karri, J. Rajendran, K. Rosenfeld, Trustworthy hardware: Trojan detection and design-for-trust challenges. Computer **44**(7), 66–74 (2011). doi:10.1109/MC.2010.369
61. Mohammad Tehranipoor, Farinaz Koushanfar, A survey of hardware Trojan taxonomy and detection. IEEE Des. Test Comput. **27**(1), 10–25 (2010). doi:10.1109/MDT.2010.7
62. Ramesh Karri, Jeyavijayan Rajendran, Kurt Rosenfeld, Mohammad Tehranipoor, Trustworthy hardware: identifying and classifying hardware Trojans. Computer **43**(10), 39–46 (2010). doi:10.1109/MC.2010.299
63. M. Subhasish, H.-S. Philip Wong, S. Wong (2015). http://spectrum.ieee.org/semiconductors/design/stopping-hardware-trojans-in-their-tracks
64. https://www.sbir.gov/sbirsearch/detail/399376—Commodity Goods Counterfeit Detection
65. https://www.sbir.gov/sbirsearch/detail/413453—Electronic Component Fingerprinting to Determine Manufacturing Origin
66. https://www.sbir.gov/sbirsearch/detail/561382—Development and Verification Tools/Processes for ASICs and FPGAs
67. https://www.sbir.gov/sbirsearch/detail/736721—Rapid Non-destructive Detection of Advanced Counterfeit Electronic Material
68. https://www.sbir.gov/sbirsearch/detail/825801—Detecting Counterfeit, Sub-standard, Non conforming, and Improperly Processed Material
69. https://www.sbir.gov/sbirsearch/detail/870419—Integrated Circuit Authentication and Reli ability Tool and Techniques
70. https://www.sbir.gov/sbirsearch/detail/870417—Multi-Attribute Circuit Authentication and Reliability Techniques
71. https://www.sbir.gov/sbirsearch/detail/870269—Rapid and Reliable Identification of Counterfeit Electronic Components

Part II
Solutions

Manufacturing Solutions

Adrian Evans, Said Hamdioui and Ben Kaczer

Abstract The continued scaling of CMOS transistors has been the enabler of faster, cheaper, and denser ICs and electronics. However, as the scaling is slowly coming to its end, many challenges emerge, including higher static power, high manufacturing cost, and more important, reduced reliability. The latter is mainly due to process or time-zero variation (i.e., process variations) or time-dependent variations (either related to temporal/aging variations such as SILC, BTI, HCD, or to environmental variations such as radiations). This chapter provides a broad overview of the latest techniques that are being used to mitigate these reliability challenges in the latest technology nodes. In the first part of this chapter, we present some of the techniques that are being used to manage process variation, including both static and dynamic techniques. These techniques span the entire range from improved layout rules to dynamic voltage scaling all the way to techniques implemented in application software. In the second part of this chapter, a review of these aging effects is presented including SILC, BTI, HCD, and self-heating effects, as well as the latest research on how they can be mitigated. In the final section, we investigate radiation-induced upsets and how they impact the latest technology nodes including FinFET and SOI technologies.

A. Evans (✉)
IROC Technologies, 2 Square Roger Genin, 38000 Grenoble, France
e-mail: adrian.evans@iroctech.com

S. Hamdioui
Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands
e-mail: S.Hamdioui@tudelft.nl

B. Kaczer
Imec, Kapeldreef 75, 3001 Leuven, Belgium
e-mail: ben.kaczer@imec.be

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_4

1 Introduction

In the previous chapters, many of the challenges in the design of reliable nanoscale devices have been described. Many of these challenges such as manufacturing faults and transient faults have existed for many generations of CMOS, and a large body of knowledge around design for test, redundancy, and hardening techniques has developed. Today, advances in CMOS are less the result of scaling and increasingly the result of innovation in terms of process, materials, and new types of transistors. Combined with the fact that the dimensions of transistors are approaching the atomic scale, variability is increasing and the total number of transistors per die is often in the billions, new reliability challenges are emerging.

In this chapter, we address new approaches for addressing reliability threats, with a focus on the process level. In the first section, we explore new approaches for managing increased process variation. The following section discusses how transistor degradation due to gate oxide breakdown, BCI, HCI, and self-heating effects can be mitigated at the process level. Next is a section that discusses the trends in radiation sensitivity in the most recent CMOS nodes including how this impacts the design of radiation-hardened cells. The total power drawn by large integrated circuits can be very significant; thus IR drop is a real problem. The final section in the chapter discusses this challenge and how voltage droop can be managed.

The focus of this chapter is how the reliability challenges in advanced CMOS devices can be managed, primarily at the materials, process, and technology level. Subsequent chapters will investigate higher level approaches, at the micro-architectural and architectural level.

2 Mitigation of Process Variation

Process variation affects the speed and power consumption of circuits. Some circuits may fail to meet the intended speed, if process variation is not taken into account during design. This results in a low yield. Conventionally, guard bands are built into the design to account for process variation. With the ongoing shrinking of CMOS technology, the effects of process variation become more and more pronounced. Because of this, a guard-banded design leads to an increased penalty in terms of area and power consumption.

An often performed technique to remedy the effects of process variation is *speed binning*. With speed binning, chips are tested extensively after production in order to find their maximum clock speed and classify them. The faster chips are then sold at higher prices, while the slower ones at lower prices. Therefore, the extensive testing pays off. These days, however, processors have shifted from single core to multi-core. Therefore, an increase in clock speed has become less interesting and instead more cores are preferred. Because of this, speed binning is becoming less profitable.

The drawbacks with guard-banded design in terms of power and area, and the decrease in effectiveness of speed binning have led to the development of techniques to mitigate process variation. Thanks to these techniques, high yield is guaranteed without adding excessive guard bands.

2.1 Classification

Figure 1 shows a high-level classification of process variation mitigation schemes. As it can be seen, the schemes can be divided into static and dynamic ones. Static schemes are used during the design or the manufacturing of the chip; they can even be tuned once before deploying the chip in the application. However, these schemes cannot be tuned at runtime and during the lifetime of the chip, which is the case for dynamic schemes. Such schemes are based on monitoring the circuit's behavior at runtime and taking action when necessary, in order to prevent errors in the circuit.

2.2 Static Schemes

Figure 2 shows the classification of the most common static process variation mitigation schemes. As it can be seen, schemes can be applied either during the process phase or during the design phase; they are discussed next.

2.3 Process Schemes

The process schemes include all techniques applied during fabrication to minimize process variation; these may be related to the used materials or to techniques to increase the resolution of structures printed on silicon, called Resolution Enhancement Techniques (RETs).

Material: The used materials for the production of semiconductor devices are constantly evaluated and improved, especially for emerging devices and new materials. A good example of a newly introduced material that helped mitigating process variation is high-κ dielectrics in the 45 nm technology [1]. The high-κ

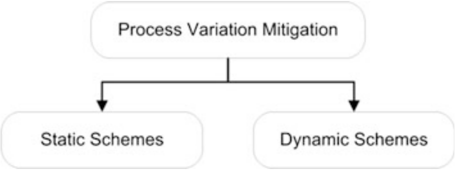


Fig. 1 Process variation mitigation classification

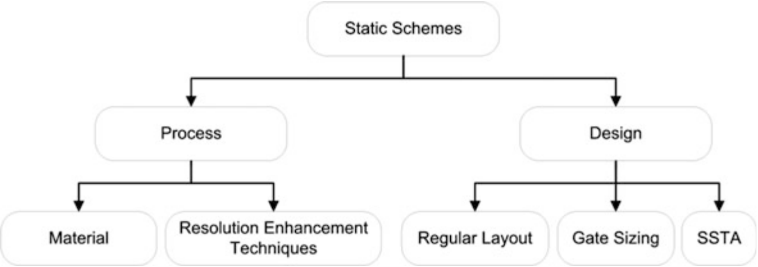


Fig. 2 Static process variation mitigation schemes classification

dielectrics are used to replace the conventional silicon dioxide (SiO_2) that was used for the gate oxide material. The thickness of the gate oxide used to steadily decrease as transistors decreased in size, until leakage currents became a concern, and the scaling slowed down. With the introduction of high-κ dielectrics for the gate oxide, the gate leakage is reduced, making further gate oxide scaling possible. This scaling has a positive effect on random variations due to Random Dopant Fluctuation (RDF), because matching of transistors improves when gate oxide thickness decreases [2].

Resolution Enhancement Techniques: The current lithographic process for making chips utilizes ultraviolet light with a wavelength of 193 nm. As the dimensions of current nanometre-scale technology are only a fraction of this wavelength, it becomes difficult to produce the requested patterns. This happens due to diffraction effects, which defocus the patterns printed on silicon. The diffraction effects have resulted in the introduction of several Resolution Enhancement Techniques (RETs), which counteract the diffraction effects and, thus, increase the resolution of the lithography step. Thanks to the increased resolution, the process variation is reduced as well.

A common RET is to use *phase-shift masks* [3]. Phase-shift masks alter the phase of the light passing through certain areas of the mask, which in turn changes the way the light is diffracted and, therefore, the defocusing effect is reduced.

Another RET is Optical-Proximity Correction (OPC) [4–6]. OPC pre-distorts the mask data in order to compensate for image errors due to diffraction effects. The pre-distortion is done by moving edges or adding extra polygons to the patterns on the mask. This results in a better printability.

Finally, *double patterning* is also a technique to increase the resolution of printed patterns [7]. With double patterning, dense patterns with a high pitch are split over two masks. The two masks then contain lower pitch structures. The dense patterns are then printed with two exposure steps. In each of the steps one of the two masks is used. The combination of the two masks then results in a higher pitch printed on silicon. This pitch is hard, if not impossible, to achieve with a single patterning process.

2.4 Design Schemes

During the design of a chip, various methods, such as regular layout styles, gate sizing, and Statistical Static Timing Analysis (SSTA), can be applied to mitigate process variation. These techniques are discussed below.

Regular Layout: Regular layout styles aim at simplifying the patterns that need to be printed on silicon. This is achieved by adding more regularity and symmetry to the design. Regular layout techniques reduce variability that occurs due to lithography distortion.

Regularity is added to the layout, for instance, by only allowing a fixed device orientation and routing direction per layer. *Ultra-regular* and *semi-regular* layouts, of which example circuits are shown in Fig. 3, were proposed in [8]. The ultra-regular layouts use a single-device orientation, constant poly pitch, and the direction of the routing is fixed. With the semi-regular layout, the width and spacing for the geometries are held as constant as possible with minor deviations allowed.

A higher regularity than the ultra-regular layouts can be achieved by using only a single or limited set of highly optimized basic building blocks and repeating these blocks, as it is the case for Via-Configurable Transistor Array (VCTA) cell proposed in [9]. The VCTA cell maximizes regularity for both devices and interconnects. The VCTA cell consists of n NMOS gates and n PMOS gates. To maximize regularity, all transistors have the same width and channel length. On top of the VCTA cell, a fixed and regular interconnect grid of parallel metal lines is placed. The functionality of the cell can be configured by connecting the transistors in the cell in a certain way. This is done by making connections between the metal lines and transistors using vias. With this, the via placement and inter-cell interconnections are the only source of irregularity in the layout of the design.

One of the advantages of regular layouts is the yield improvement due to the reduction of process variability. Another advantage is the acceleration of the time-to-market due to the lower number of basic cells and layout patterns that need to be optimized. A disadvantage of regular layouts is an increase in the area with the associated delay and power consumption. Furthermore, some regular layouts have a

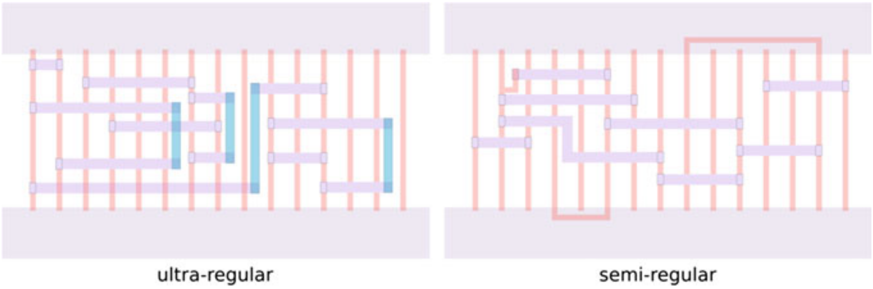


Fig. 3 4 Ultra-regular layouts and semi-regular

fixed transistor width, which may make it difficult to meet delay specifications on all paths, and will also increase power consumption.

Statistical Static Timing Analysis (SSTA): Traditionally, designers use corner analysis to ensure that the design will meet its timing specification under all cases of process variation. In corner analysis, all electrical parameters are considered to be within certain bounds. The design is valid if the circuit meets the performance constraints for all corners. For corner analysis Static Timing Analysis (STA) is used often. With STA, a circuit's timing is analyzed by adding the worst-case propagation times for the gates of a path. This analysis is only necessary if the process variations are of a systematic nature. However, in nanometre-scale technologies random variations are more dominant. This means it is unlikely that all gates in a path will show worst-case propagation times. Therefore, STA leads to an overly pessimistic design.

As an alternative to STA, statistical static timing analysis (SSTA) has been proposed [10]. In SSTA, the worst-case propagation times of gates are replaced by probability density functions (PDFs). These PDFs are then propagated through the logic network, to determine the final PDF of the propagation delay of the circuit. With the final distribution, direct insight can be obtained on the yield of the design. Therefore, high yield can be achieved without adding excessive margins.

Obviously, SSTA leads to lower die area and reduced power consumption. However, it increases the design time, due to the higher complexity of SSTA as compared to STA.

Gate Sizing: With gate sizing, an attempt is done to find optimal drive strength of gates in the circuit in order to obtain a trade-off between delay, power consumption, and area. The drive strength is set by changing the size of the transistors in the gate, which enable the design optimisation under certain constraints. For instance, the design can be optimized for area and power consumption at a minimum target speed. Different optimization algorithms have been published in literature [8–10].

Recent gate sizing techniques have started to take into account process variation as well [11, 12]. These techniques are referred to as *variation aware gate sizing*. They model process, voltage, and temperature variations using statistical methods. With these techniques power and area can be optimized at higher yield.

Gate sizing offers advantages such as reduced die area and lower power consumption. However, it complicates the design process.

2.5 Dynamic Schemes

Dynamic mitigation schemes monitor the circuit's behavior online (in field) and when necessary actions are taken to prevent timing errors. Figure 4 shows the classification of the most common dynamic mitigation schemes. Note that dynamic schemes can be applied at the hardware or at the software level; both approaches are discussed below.

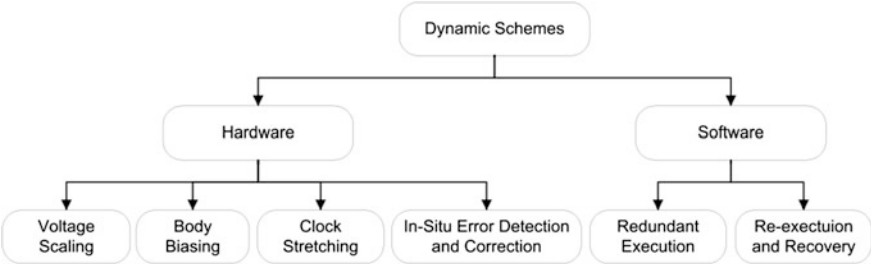


Fig. 4 Dynamic process variation mitigation schemes classification

2.6 Hardware-Based Schemes

As can be seen in Fig. 4, hardware techniques to mitigate process variation include voltage scaling, body biasing, adaptive clocking, and error detection and correction.

Dynamic Voltage Scaling (***DVS***): this is a technique where the supply voltage is scaled down in order to limit the dynamic power consumption. Most modern processors dynamically change the clock frequency based on throughput requirements in order to save energy. This tuning of the clock frequency can happen in conjunction with voltage scaling, as a lower frequency requires a lower voltage in order to still meet the timing. Thanks to this scaling of the supply voltage, even more power is saved compared to only lowering the clock speed.

Conventional DVS techniques require enough supply voltage margin to cover process variations, which results in wasted energy. Therefore, variation aware DVS is proposed. With this technique on-chip monitors are added to the circuit to provide feedback on the process variation in the circuit. Based on this feedback, the supply voltage can be adjusted to the near minimum level needed to run without errors. Early papers are based on critical path replication and monitoring this replica. For instance, in [13] the critical path of the system is replicated with a ring oscillator. Based on the measured frequency of the ring oscillator, it is determined if the supply voltage can be lowered or should be increased. Due to the growing on-die process variation in nanometre-scale technologies, using a single reference structure is no longer feasible, because in this case extra margin is necessary. Furthermore, it is becoming more and more difficult to select a unique critical path across all conditions. A technique to emulate the actual critical path under different process and parasitic conditions was described in [14]. Thanks to the close tracking of the actual critical path, the supply voltage can be scaled down further. An in situ delay monitoring technique was proposed in [15]. For this, pre-error flip-flops are used; they are capable of detecting late data transitions. The power supply is then scaled based on the rate of errors.

An advantage of DVS is a reduction in power consumption as the supply voltage is closer to its minimum value. A disadvantage of DVS is that it is mainly suitable for global variations, because it is difficult to find a unique critical path. To better account for local variations, more reference circuits for performance monitoring are needed on the die. To account even better for local variations, the circuit should be split up into multiple sub-circuits with separate supply voltages, so each sub-circuit is supplied with its minimum operation voltage. This results in even higher area overhead.

Body Biasing: it is a technique that allows to change the threshold voltage of a transistor. With body biasing the transistor's *body effect* is utilized; it refers to the dependence of the threshold voltage on the voltage difference between the source and body of the transistor. Normally, the NMOS transistor's body is connected to ground and for a PMOS transistor's body to V_{DD}. By applying different voltages at the body terminals, it is possible to control the threshold voltage of the transistors. In order to do this, the body terminals of the transistors need to be connected to separate power networks instead of V_{DD} and ground. Through these power networks the body biasing voltages are then controlled.

When the threshold voltage is lowered with body biasing, it is called *forward biasing*. In this case the transistors will switch faster, making the circuit faster. This happens at the penalty of increased power consumption due to higher leakage. It is also possible to increase the threshold voltage; this is referred to as *backward biasing*. This makes the circuit less leaky, which leads to a lower power consumption at the cost of a slower circuit.

Body biasing can be used to mitigate process variation. On slow circuits forward biasing is performed in order to make them faster. On fast circuits, which suffer from higher leakage, backward biasing is performed. The required body biasing voltages can be applied with the use of *on-chip* sources, such as power regulators. Just like with DVS, on-chip monitors are added to the die that measure a test structure to determine the process variation. In [16, 17] a ring oscillator is used to measure the process variation in a circuit. Based on the ring-oscillator measurements the power regulators generate appropriate biasing voltages to mitigate the effects of process variation. Note that if only one test structure is measured, only global, systematic variations can be mitigated and still a margin is necessary to account for within-die variations. Accounting for within-die variations requires special attention; e.g., in [17], the authors proposed to divide the circuit into multiple sub-circuits with separate body bias networks. By monitoring ring oscillators close to the sub-circuits, each sub-circuit can then be supplied with unique biasing voltage. This way the within-die variation is compensated to a certain extent, which improves the frequency and the leakage of the circuit even more.

An advantage of body biasing is a reduction in power consumption, as the leakage of chips from the fast corner is reduced. Just like with DVS, a disadvantage of body biasing is that it is mainly suitable to compensate global variations. To better account for local variations, more test structures on the die are needed and the circuit needs to be split up into sub-circuits that each has their own body bias network. This results in a higher area overhead.

Clock Stretching: under process variation, some circuits may fail to meet timing. Often, critical paths that exceed the maximum delay are responsible for this; critical paths have the least amount of timing margin and, therefore, are the first to fail. As a solution for this, clock stretching has been proposed. The idea is to *stretch* the clock when a critical path is activated. This gives the path more time to finish propagation and, therefore, timing errors are avoided. The concept of clock stretching is illustrated in Fig. 5. As can be seen, in cycle 2 the computation time, which indicates the highest propagation time of activated paths in the circuits, exceeds the normal clock period. Therefore, the clock is stretched to two cycles in order to avoid timing errors.

One of the challenges with clock stretching is predicting when a critical path is activated. One way to realize this is to use a *pre-decoder* as proposed in [18]; the pre-decoder has as input, the input vector to the logic. Based on this input vector, the pre-decoder predicts critical path activation in the circuit. When a critical path is activated, a signal is asserted to stretch the clock. An example of an adder with a pre-decoder to enable clock stretching is shown in Fig. 6. As can be seen, the pre-decoder relies on some adder's inputs to insert the clock stretching when needed.

One of the challenges with using a pre-decoder to predict critical path activation is the area overhead and the additional wiring, especially for big circuits. This will most likely make the pre-decoder relatively big. An alternative to the pre-decoder is the CRISTA design [19]; CRISTA isolates the critical paths and makes their activation predictable. This is achieved by partitioning and synthesizing the circuit into several separate logic blocks. These blocks contain each a primary input, indicating if the block is active or idle. The design is synthesized in such a way that only some of these blocks contain critical paths. With the use of the active/idle signal, it is then easy to predict if critical paths are sensitized. Figure 7 shows an example path delay distribution for a design with CRISTA. The targeted delay of one cycle is indicated. It can be seen that a set of paths exceeds this delay. CRISTA makes the activation of these paths is predictable. When one of these paths is activated, clock stretching is performed, so there is enough time for the circuit to finish propagation. It can also be seen that the other set of paths has a lot of slack, which provides resilience against process variation.

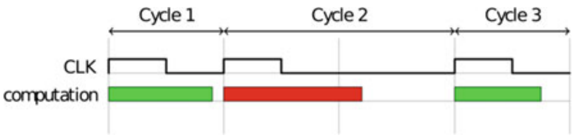


Fig. 5 Illustration of clock stretching

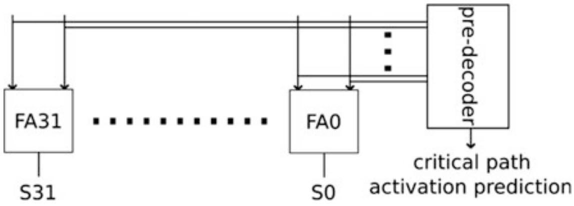


Fig. 6 Adder circuit with pre-decoder for clock stretching

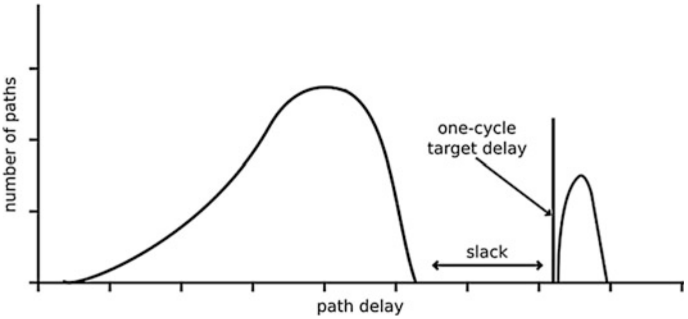


Fig. 7 Path delay distribution required for CRISTA

A disadvantage of clock stretching is the speed degradation of the circuit. This happens due to the fact that sometimes the clock period is longer. Another disadvantage is that the area overhead to enable prediction of the activation of critical path can become high and, therefore, not all circuits are suitable for such as scheme.

In situ error Detection and Correction: with *in situ error detection*, timing errors are detected. This is done by checking for late transitions at data inputs of flip-flops. Typically, flip-flops are augmented with a latch or a second delayed clock input in order to check for late transitions. Usually, these techniques are applied in pipeline circuits. One of the earliest works on in situ error detection is Razor [20], for which an example of a pipeline stage is shown in Fig. 8a. As can be seen, each flip-flop is augmented with a *shadow latch*, which is controlled by a delayed clock. A timing diagram to illustrate how Razor works is shown in Fig. 8b. In the first clock cycle, logic stage L1 meets the normal timing. In the second cycle, however, logic stage L1 exceeds the intended delay. Therefore, the data (instr 2) is not captured by the main flip-flop at clock cycle 3. The shadow latch does capture this data, since it operates with a delayed clock. Because the data stored in the main flip-flop and the shadow latch differ, the error signal is raised and the preceding pipeline stages are stalled. After this, the valid data is restored in the fourth cycle. Therefore, the error is corrected with a penalty of one clock cycle delay.

Razor corrects timing errors in the circuit at the penalty of one clock cycle delay. There are also techniques that mask the timing error, e.g., by delaying the arrival of the correct data to the next pipeline stage. Authors in [21] proposed the TIMBER flip-flop; a flip-flop that has a delayed clock input to resample data input for any timing errors. In case of a timing error, the output of the flip-flop is updated with the correct value, which is then propagated to the next stage of the pipeline. In this case, time is borrowed from the succeeding pipeline stage.

In situ error detection can be used to mitigate process variation. Timing errors that occur due to critical paths affected by process variation can be detected and corrected. Hence, fault-free operation of the circuit can be achieved without adding a lot of margins to the design.

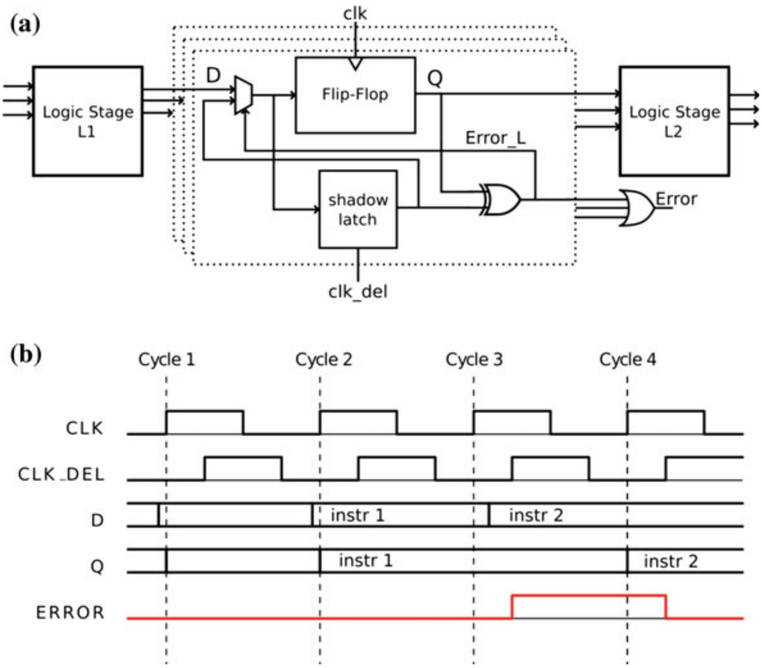


Fig. 8 Pipeline augmented with Razor

An advantage of in situ error detection is its capability to compensate local variation, besides global variation. This is due to the fact that all flip-flops or only flip-flops ending at critical paths are augmented with in situ error detection. Because of this, timing errors on critical paths that occur due to local variation are detected. A disadvantage of in situ error detection is a possible decrease in throughput, due to the correction. Another disadvantage is a high area overhead due to the fact that most flip-flops need to be augmented with error detection and also the control logic that is needed to handle the errors.

2.7 *Software-Based Schemes*

In addition to mitigating process variation at the hardware level, it is also possible to mitigate process variation at the software level. As technology scales further, reliability becomes a more challenging design factor. This is due to, for example, increased aging effects and increased vulnerability to soft errors. Software methods are being developed to detect errors in order to be able to guarantee dependable computing. A technique that can be employed is *redundant execution* [22], where critical portions of the software are run redundantly on multiple cores. The outputs are then compared to see if any errors are introduced. Another method is *Re-execution* *and Recovery* [23], which provides resilience by re-executing portions of the application that have been detected as being corrupted. These software techniques can also be applied to mitigate process variation, besides mitigating aging and soft errors.

3 Mitigation of Transistor Aging

As device dimensions are downscaled in the relentless effort to keep with Moore's law, maintaining gate control and suppression of short-channel effects requires the introduction of new FET architectures. The semiconductor industry has already moved to FinFET or Fully Depleted Silicon-On-Insulator devices. These are expected to be superseded by nanowire devices with the gate fully wrapped around the channel. At the same time, high-mobility substrate materials, such as Ge, SiGe, and IIIV compounds, are being investigated to accelerate device operation.

As smaller devices, more complex device architectures, and new materials are being introduced, the reliability margins continue to shrink. In many cases, the reliability margin assuming continuous operation at elevated temperature (Fig. 9a) may be no longer sufficient [24]. Below, the main degradation processes affecting FET devices are first discussed (Sects. 3.1–3.4), along with their overall trends with technology scaling. The root cause of gate oxide degradation—generation and charging of interface and bulk gate oxide traps—is common to all the main degradation mechanisms. The technological means of reducing both interface and bulk traps are therefore discussed in Sects. 3.5 and 3.6.

Since devices in realistic digital circuits typically operate with a series of high and low signals, while the supply voltage V_{DD} changes as, e.g., the "turbo" and the "sleep" modes are enabled, assuming more realistic workloads will result in a more realistic prediction of the mean degradation (Fig. 9b), thus regaining some of the projected reliability margin. In addition to that, correct understanding of the effect of a degraded device on the surrounding circuit will allow to better mitigate aging-related issues already during the design phase. Examples of this are given throughout the text.

On the other hand, only a handful of defects will be present in the gate oxide of each deeply scaled device. This will cause an increase of the so-called time-dependent, or dynamic, device-to-device variability. The same workload will result in a device-to-device *distribution* of degradations (Fig. 9c). The time-dependent variability is discussed in Sect. 4.

3.1 Stress-Induced Leakage Current and Gate Oxide Breakdown

Generation of conducting defects in the bulk of the gate dielectric during device operation leads to an increase in gate current (leakage). This phenomenon is

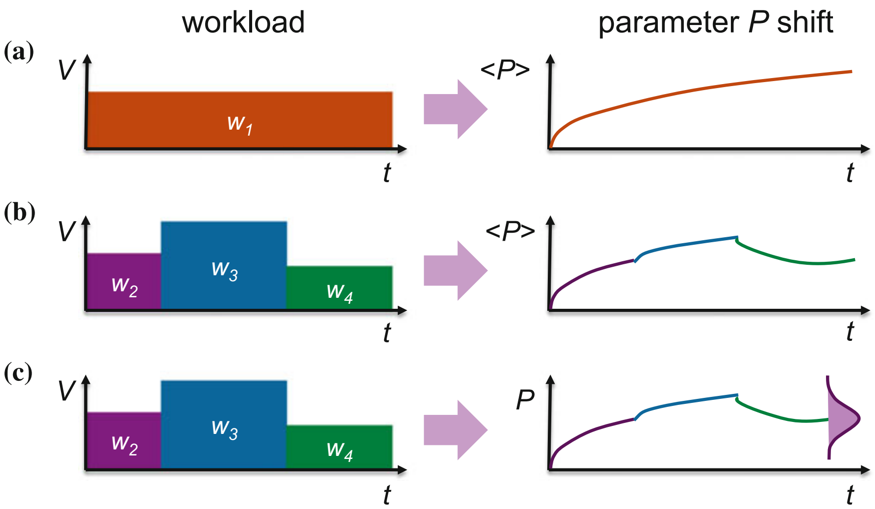


Fig. 9 Approaches to device degradation projections. **a** "Conventional" projection: mean degradation is estimated from the "worst-case" constant (i.e., DC) stress at maximum V_{DD} (workload w_1) applied for the entire duration of application lifetime. **b** More realistic workloads w_n (with different voltages, frequencies, duty cycles, temperatures, etc.) result in better end-of-lifetime *mean* degradation prediction. **c** In reality, a *distribution* of device-to-device degradation needs to be considered

therefore termed Stress-Induced Leakage Current (SILC). SILC can potentially partially eliminate gate current leakage reduction gained by the introduction of high-κ gate dielectrics [25]. At sufficiently high density the newly generated defects will form a *percolation* path between the gate and the body of the FET device, resulting in so-called Soft Breakdown (SBD). The current through a formed SBD path is typically a strongly superlinear function of gate bias and of the order of ~μA at 1 V.

The breakdown path can further progressively wear out and when a sufficient local current is reached, a runaway defect generation at the breakdown spot will lead to a so-called Hard Breakdown (HBD). HBD current–voltage characteristic is near-ohmic, with typical values of 1–10 kΩ.

All of the above processes, often called Time-Dependent Dielectric Breakdown (TDDB), are accelerated by gate voltage, current, and temperature [26]. The continuing voltage and power reduction is therefore generally beneficial for increasing and thus postponing time to soft breakdown t_{SBD}. Oxide downscaling also affects PFET t_{SBD} more than NFET, because the gate current in PFETs is due to direct tunneling, while in NFETs it is due to Fowler–Nordheim tunneling—a leakage mechanism less sensitive to thickness variations [27]. The employment of gate metals with more midgap work functions (see also Sect. 3.2) is also beneficial in this sense [27].

In gate stacks with high-κ dielectrics, "Alternating Current" (AC) TDDB is frequency dependent, with low frequencies apparently decreasing t_{SBD} [28, 29]. This appears to be related to bulk high-κ traps, in particular their charging and discharging during the AC stress [30].

The post-SBD progressive wear-out is controlled by the voltage across the breakdown path and the current running through it. The SBD wear-out progress will be therefore slowed down if the stress bias is supplied from a non-ideal "soft" voltage source capable of providing only limited current, such as the preceding transistor stage [31].

If SBD does occur in a FET, the FET drain current characteristic will be typically little affected (Fig. 10). This is because of the limited current of the SBD spot [32]. FET width or the number of fins can be also upsized during design to compensate for the breakdown current. Sufficiently wide devices can then compensate even for HBD [33].

Gate oxide defect generation proceeds in parallel at different locations of all stressed FET gates and multiple SBD formation at different parts of the circuit [35] or even a combination of SBDs and HBDs is possible [36]. With proper device sizing, multiple SBD breakdowns will only affect power consumption. The statistics of time-to-nth breakdown has been developed [35, 37] allowing to reclaim some reliability margin.

3.2 *Bias Temperature Instability*

Bias Temperature Instability (BTI) is caused by charging of pre-existing and generated defects in the bulk and at the interfaces of the gate dielectric [38]. It is accelerated by gate oxide electric field and temperature. The issue in pFET devices, so-called Negative BTI (NBTI), was exacerbated by the introduction of nitrogen into SiO_2 gate dielectric [39, 40]. The complementary mechanism in nFET devices, Positive BTI (PBTI), became a significant issue with the introduction of high-κ gate dielectrics. As the semiconductor industry moves to FinFET and FDSOI devices, channel doping can be reduced due to better channel control, resulting in the reduction of depletion charge. As a consequence, the gate work function can be adjusted toward Si midgap and the gate oxide field can be reduced at given V_{DD} and threshold voltage V_{th} with respect to planar devices [27]. Further reduction in depletion charge will come from reducing the fin width below the depletion width [41]. For future technology nodes, the electric field in the oxide is expected to increase, as the oxide thickness is reduced faster than V_{DD} to help maintain channel control. One exception is the so-called junctionless FETs [42], which operate in partial depletion or in accumulation [43]. Such devices have high flat-band voltage, resulting in low field and hence low degradation during operation [44].

AC BTI results in significantly lower degradation than the equivalent fully on "DC" BTI stress. This is because of so-called relaxation of BTI, due to discharging

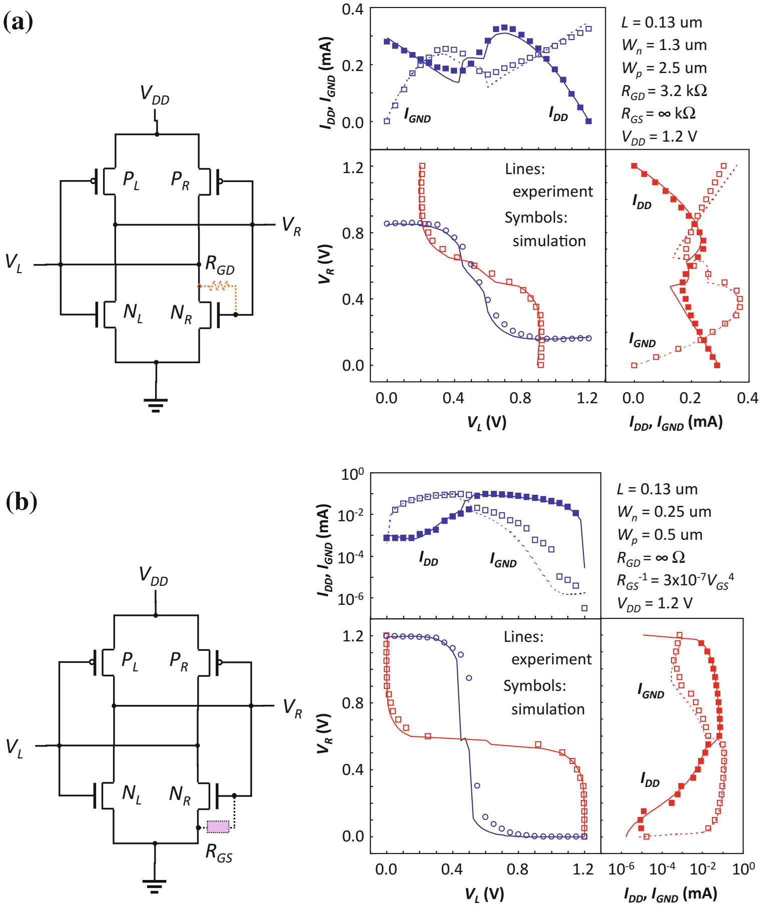


Fig. 10 **a** An SRAM cell with wide FETs can compensate even a hard breakdown. SRAM transfer characteristics after hard FET N_R drain-side BD are well reproduced by a simulation assuming R_{GD} = 3.2 kΩ. Two stable points can still be discerned in the butterfly plot. **b** Narrow-FET SRAM characteristics after *soft* source-side BD in FET N_R are well reproduced by simulation assuming a nonlinear, weakly conducting path. The cell's characteristics are not strongly affected after SBD [34]

of defects. At very low frequencies (<100 Hz), the relaxation also naturally explains frequency dependence of AC BTI [45–47]. At intermediate and high frequencies (~GHz), there is presently disagreement in the literature [48–50]. Part of the confusion seems to arise from experimental issues at high frequencies. When high-frequency signal integrity issues are correctly considered, NBTI is decreasing at high frequencies due to multistate nature of the involved traps [51], while PBTI is frequency independent [52].

3.3 Hot Carrier Degradation

When a FET is biased in inversion and a bias is also applied at the drain, the channel carriers arriving at the drain will not be in equilibrium with the semiconductor lattice. The "hot" carriers at the high energy tail of the energy distribution will be then responsible for (localized) generation of interface states (through hydrogen depassivation) and charging of the bulk states in the dielectric, either directly, or through the carriers of opposite polarity generated simultaneously through impact ionization. This set of processes is termed Hot Carrier Degradation (HCD). Note that BTI degradation due to "cold" carriers can still take place at the source side. Finally, heat will be generated as energy from the hot carriers is transferred into the semiconductor lattice, resulting in the so-called Self-Heating Effect (SHE) and accelerating some of the degradation above degradation mechanisms. The symptoms include drain current, transconductivity and subthreshold degradation, and threshold voltage shift.

Generally, the lateral electric field in the channel, particularly at the drain, will have a strong impact on the energy distribution function and hence on the above degradation processes. Therefore, even though the supply voltages V_{DD} and hence the maximum drain voltages are gradually decreasing, this degradation mechanism becomes more pronounced as the gate length is reduced [53]. The gate oxide electric field also increases as the gate oxide is scaled down. Hot carrier degradation is presently flagged as most critical reliability concern in the upcoming technology nodes.

Junction optimization to lower the electric field at the drain is therefore generally mandatory to alleviate the impact of hot carrier degradation [54, 55]. The decreased oxide electric field in junctionless FETs can decrease HDC effects [56].

The fin width in FinFET devices is a critical parameter. Both reduction and acceleration of HCD with the fin width have been reported [27, 57–60]. The disparate results are likely due to the complex dependence of the involved mechanisms. As the fin width changes, so does the threshold voltage and the electric field profile in the fin [59], junction profiles, and the amount of heat retained in the fin due to SHE [27]. This will result in the energy distribution function varying strongly with the fin width [59]. Furthermore, the fraction of hot carriers impinging on the gate oxide will change with changing fin width as well [27].

HCD is a cumulative process and AC HCD does not seem not frequency dependent [52].

3.4 Self-heating Effect

When the FET device is operating at $V_{\mathrm{D}} = V_{\mathrm{DD}}$, considerable power $I_{\mathrm{D}} * V_{\mathrm{DD}}$ is dissipated in the device. In planar devices, the excess heat is primarily dispersed into the silicon substrate (bulk Si thermal conductivity $\sim 148\ \mathrm{W\ K^{-1}\ m^{-1}}$). The remnant heat raises the device body temperature above that of the chip. This is called the Self-Heating Effect (SHE). Although not strictly a degradation mechanism of its own, SHE can accelerate other degradation processes in the FET.

As device geometry changes from planar to multi-gate, the relative thermal contact of the device with the silicon substrate decreases. Heat has to escape into the gate through the gate oxide (bulk SiO_2 thermal conductivity ~1.40 W K^{-1} m^{-1}) and the source and drain contacts. This phenomenon is further amplified if Silicon-On-Insulator (SOI) technology is used (Fig. 11) [57].

New high-mobility materials presently under consideration may have lower thermal conductivity than Si (Ge bulk thermal conductivity 58 W K^{-1} m^{-1}, GaAs bulk thermal conductivity 58 W K^{-1} m^{-1}). Thermal conductivity also decreases at elevated temperatures and with dopant concentration (the latter is fortunately reduced in modern devices). In deeply scaled devices, the impact of material interfaces is amplified as they will scatter the heat-carrying phonons, resulting in severely reduced thermal conductance values (fractions of the bulk values) [61].

Temperature generally accelerates single-carrier interface state depassivation and bulk charging; it, however, also reduces the mean-free path of the hot carriers, thus lowering their average energy. However, the tail of the energy distribution can expand with temperature, accelerating one type of interface bond depassivation mechanisms [62]. Also the BTI degradation taking place at the source will be accelerated. Separation of the concomitant degradation mechanisms for proper lifetime projection is therefore a considerable challenge.

SHE can be generally alleviated by improving the heat escape paths. In FinFET devices, SHE can be reduced by sufficient spacing of the fins [63, 64]. Reduction of buried oxide in SOI devices is also highly beneficial [65]. Finally, assuming the actual workload already during will result in better estimate of the dissipated heat, actual temperature, defect generation and charging rates, and hence better lifetime estimation.

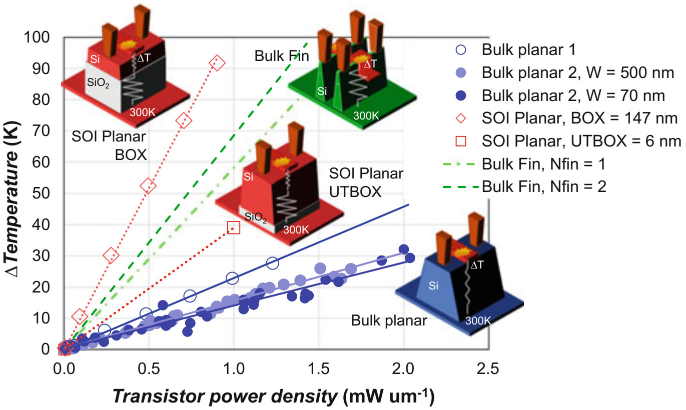


Fig. 11 Across-technology plot based on measurements (*dots*) and simulations (*dotted lines*) for bulk planar, SOI planar and bulk FinFETs showing the local temperature rise in the FET as a function of the power density [57]

3.5 Root Cause 1: Interface Traps

Dangling silicon bonds at the Si/SiO_2 interface act as energy states within the Si band gap. In standard CMOS process flow, these bonds are passivated with hydrogen during chip fabrication. When the chip and the devices are biased during use in the field, especially during negative gate bias, electrically active defect states are again generated at the Si/SiO_2 interface by stripping (depassivating) the bonds of hydrogen by interaction with channel holes [66]. Interface state generation is also a crucial component of HCD, especially in conventional devices with SiO_2 gate oxide [55]. The bond dissociation mechanism during HCD is relatively complex, and can be triggered by a single, sufficiently energetic carrier, or through multiple vibrational excitations (MVE) of the bond by multiple, lower energy carriers [67].

In standard planar devices the Si surface has (100) orientation. In FinFET devices the Si fin sidewalls have (110) orientation with a higher density of Si dangling bonds. Their depassivation can therefore contribute more to NBTI [68]. The contribution of side-wall interface states is also reduced when fins are rotated 45° around the vertical axis [58, 69], as the side-wall orientation of these devices changes from (110) back to lower density (100).

Since the Scanning Tunneling Microscopy experiments on passivated Si surfaces [70, 71], it has been established that passivation of the Si/SiO_2 interface by deuterium will result in stronger bonds less susceptible to desorption by hot electrons [72]. In general, passivating these bonds with other elements with higher atomic mass, such as fluorine, has been reported in the literature to reduce the interface defect state generation [73, 74]. Higher atomic mass is presumed to change the vibrational frequencies associated with the dangling bond and better coupling with phonon modes in the Si substrate and thus faster “cooling” of vibrations [75].

Deuterium passivation has been shown to be beneficial for reducing interface state generation due to HCD [76] and NBTI [77], although in the latter case interface state generation may not be the main component in high-κ based dielectrics (see next section). Fluorine passivation has been reported beneficial for HCD [78] and NBTI [79, 80], although the effect seems to be strongly dependent on the F amount and processing conditions. (Low-voltage) SILC is also suppressed by F implantation resulting in lower gate current, although it does not influence defect generation efficiency [26].

3.6 Root Cause 2: Oxide Bulk Traps

Charge trapping into pre-existing defects appears to be the main contributor to both NBTI [81] and PBTI. Ubiquitous hydrogen has been reported as the main source of hole traps in SiO_2 [82, 83]. It is thought to be for both multistate switching traps and as a precursor for permanent hole trapping [81].

The contribution of bulk defect increases as advanced materials, such as high-κ gate dielectrics (responsible for rise of PBTI due to electron trapping) and Ge and IIIV substrates are introduced [84]. A significant progress in understanding the reduction of both PBTI and NBTI has been achieved with the "energy-alignment" model (Fig. 12) [85, 86]. In HfO_2 high-κ gate dielectrics, PBTI can be reduced by incorporating rare-earth elements or even nitrogen, which redistribute charge around oxygen vacancies and shift the electron trap energies toward the HfO_2 conduction band, thus misaligning them with the channel electrons (Fig. 12a) [85, 87–89]. In contrast, equivalent "defect level shifting mechanism" has not been known for NBTI (Recent work claims adjustment of hole traps by dopants [40]). However, the introduction of Ge, a high-carrier-mobility semiconductor, results in shifting the inversion channel hole energy level upward (Fig. 12b). This again results in misaligning the channel holes with the defects in the dielectric, resulting in sizable reduction in SiGe pFET NBTI degradation. Recently, shifting the trap levels in the high-κ layer has been also achieved by engineering a dipole at the interface with the SiO_2 interfacial layer [90]. Figure 13 illustrates that misaligning defect levels (Scenario 2) is significantly more efficient at low (operating) gate overdrives ($V_{ov} = V_{dd} - V_{th}$) than reducing defect density "en bloc" (Scenario 1), which also takes place as the gate oxide thickness is reduced.

"Passivating" the oxygen vacancies by optimizing nitrogen incorporation is also shown to reduce SILC and TDDB [91] and HCD [92]. Reduction of bulk high-κ defects by higher PDA temperature as well as Zr incorporation and high-κ/metal gate interface roughness also reduce SILC [26]. Furthermore, discharging high-κ traps during stressing, e.g., with bipolar AC stress, appears to lead to SILC reduction [28, 30].

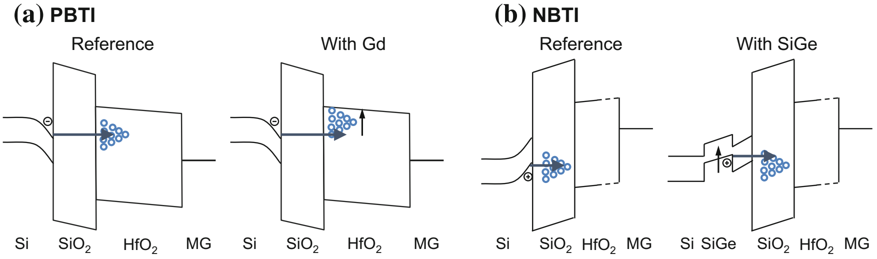


Fig. 12 A schematic illustrating the reduction of charge trapping by decoupling defect and channel energy levels **a** in nFETs (PBTI), by introducing "doping" elements into the high-κ dielectric layer, and **b** in pFETs (NBTI), by introducing low-bandgap Ge into the substrate

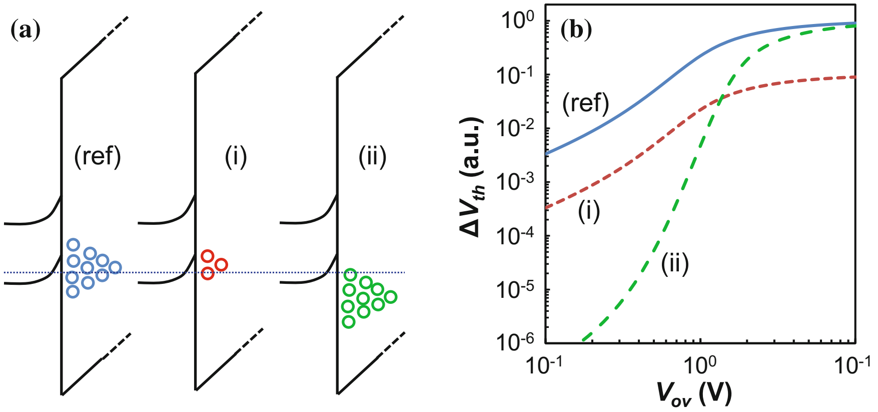


Fig. 13 **a** Charge trapping is suppressed by reducing the dielectric defect density (*i*), or by carrier/defect energy decoupling (*ii*), with respect to the reference case (*ref*). **b** Calculated ΔV_{th} assuming a 10× defect density reduction by process improvement (*i*), or the same defect density of states with mean shifted by 0.5eV (*ii*). The latter case clearly reduces BTI significantly more at low operating V_{ov} [84]

3.7 Mitigation of RTN and Time-Dependent Variability

In Sect. 3 we have discussed the origins of several aging mechanisms and possible remedies to lower their *average* or *mean* impact on the device. As device dimensions are aggressively reduced, all aging mechanisms become distributed. This *time-dependent variability* is discussed in this section.

The gate oxide thickness was the first dimension of deeply scaled FETs to reach nm length scale. The formation of the percolation path during TDDB is a stochastic process and the time-to-first SBD is described by the Weibull distribution with the mean $\langle t_{SBD} \rangle$ and scale parameter β, also known as the Weibull "slope". The variance of the distribution is reciprocal with β (smaller β results in a large distribution variance) [93].

One of the signatures of the conducting path formation process is that the variance of time-to-SBD distribution strongly increases as the *physical* oxide thickness is scaled down [93]. This is because fewer defects are needed to bridge physically thinner oxide. The introduction of high-κ dielectrics, with its increased gate oxide physical thickness, does not automatically yield reduced variance—the Weibull shape factor β is low for laminate dielectrics with, e.g., HfO_2 and ZrO_2. This could mean that either the SBD formation is controlled by the very thin SiO_2 interfacial layer or by extrinsic defects in the high-κ layer [94]. The latter case underlines the requirement of mastering fabrication of high-κ layers with low defect density and free from other imperfections, such as sharp fin edges (Fig. 14).

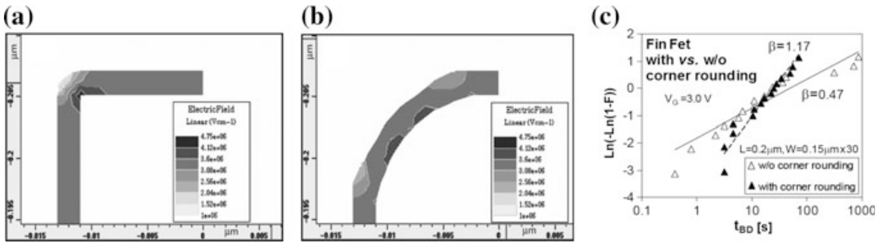


Fig. 14 Electric field distribution at the fin top corner **a** without and **b** with corner rounding. **c** Corner rounding improves (increases) the Weibull slope β [95]

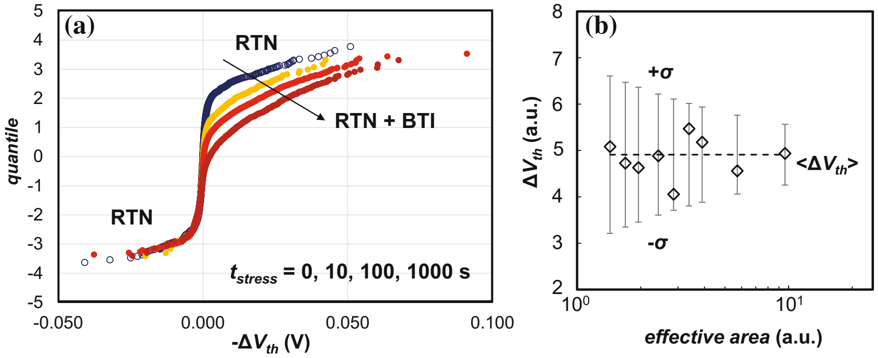


Fig. 15 **a** Tails due to RTN and due to RTN and BTI can be discerned in device-to-device distributions of ΔV_{th} of pFET devices [100]. **b** The standard deviation of device-to-device ΔV_{th} increases for smaller gate areas, as per Eqs. 1 and 2 [101]

In deeply scaled devices with typical gate areas around 1–2 × 10^3 nm^2, only 1–10 defects will be present in the gate oxide of each fin. Even at constant bias on the FET terminals, charging and discharging of individual defects will take place and result in discreet intermittent changes of the FET drain current. Such “steady-state” stochastic variations are called Random Telegraph Signal (RTS) or Random Telegraph Noise (RTN) [96]. Under certain conditions, RTN can be observed in the gate current as well [97].

If the gate is biased toward V_{dd}, the defects will become preferentially charged, contributing to BTI. The collective contribution of the charged defects to the *total* threshold voltage *shift* ΔV_{th} can be acceptably described by the so-called “Defect-Centric” or “Exponential-Poisson” (EP) distribution [98, 99] (Fig. 15a). The variance of the distribution is

$$\sigma^2_{\Delta V_{\text{th}}} = 2\eta\langle \Delta V_{\text{th}}\rangle, \tag{1}$$

where η is the average threshold voltage shift per single trapped electron or hole and $\langle \Delta V_{\mathrm{th}} \rangle$ is the mean threshold voltage shift. The means of reducing the latter have been discussed in the previous section.

The technologically important parameter η scales with oxide thickness t_{ox}, doping N_{A}, and gate area A_{G} as

$$\eta \sim \frac{t_{\mathrm{ox}}\sqrt{N_{\mathrm{A}}}}{LW}. \tag{2}$$

As can be seen from Eqs. 1 and 2, reducing η results in reduced RTN and BTI variability. From the form of Eq. 2 it is also apparent that flash memory type devices, with their minimum device sizes and large t_{ox} suffer the largest impact from individual charged defects. As in the case of as-fabricated variability, the "time-dependent" variability in logic circuit-critical devices can be reduced by increasing their gate area or fin count (Fig. 15b). Fortuitously, in logic devices η is also reduced as t_{ox} is reduced with device size to maintain control over channel, and reduced doping in the low-doped channels of FinFET and FDSOI. However, other sources of variability, such as interface states, may take over as the main sources of channel variability, resulting in η increase [102]. Since η represents the electrostatic impact of the charged traps, traps spatially deeper in the gate oxide will contribute less [103]. Since only spatially deeper gate traps are accessible in FETs with SiGe substrate, this material shows superior NBTI robustness [104].

In deeply scaled devices, HCD will also induce device-to-device variability [105], described by the EP distribution (cf. RTN and BTI var) [106]. Additional variability may arise after HCD due to enhanced generation of interface states. Due to the contribution of different defect types the total distribution will be multimodal (Fig. 16) [100, 106]. The high-σ tail of the full distribution is controlled by defects at the substrate (high η, cf. Figure 16 inset).

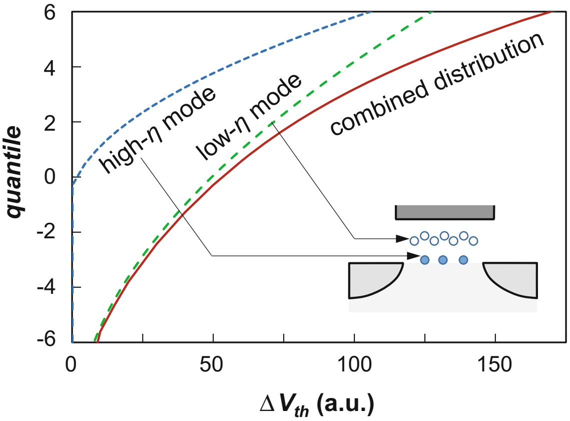


Fig. 16 Bimodal defect-centric distribution ΔV_{th} corresponding to HCD stress [106]

4 Mitigation of Radiation Effects

4.1 Introduction and Trends in Radiation Effects

Radiation effects continue to be a concern both in terrestrial and aerospace applications. The term "radiation effects" refers to a broad set of effects that occur when ionizing particles interact with silicon devices. The effects are highly dependent on the types and energies of the particles and thus on the radiative environment (e.g., terrestrial vs. space). In most cases, the device is not permanently damaged and thus these effects are often referred to as "soft errors". Some radiation effects such as gate rupture (SEGR) in power MOSFETs are destructive and thus not soft. However, in this section, the terms "radiation effects" and "soft errors" will be used interchangeably.

In terrestrial applications, the main sources of radiation that are relevant are fast neutrons, alpha particles produced from the decay of traces of unstable isotopes in the packaging materials and, for processes that contain B^{10} isotope of Boron, then thermal neutrons may also be a concern. In many applications, the latest process technologies (10 and 14 nm FinFETs) are being quickly adopted for cost, power, and density reasons. This shift is driven by the FinFET's reduced leakage current, fewer short-channel effects, and increased drain saturation current, but, as will be seen in the following sections, this new technology is also significantly less sensitive to soft errors. Indeed, many recent process technologies are immune to alpha particles and have a neutron sensitivity that is an order of magnitude lower than their planar counterparts. In this way, advances in process technology represent perhaps the most significant process level mitigation of radiation effects in terrestrial applications.

In space applications, due to longer qualification cycles, older bulk technologies are still used extensively. Here, the foremost requirement is to avoid single-event latchup (SEL). Also, in addition to the issue of single-event upsets (SEUs), space applications are also concerned about the effect that the total ionizing dose (TID), which occurs over the course of the mission. TID results in a permanent shift in transistor parameters. Although the latest FinFET and FDSOI technologies are generally not yet qualified for space applications, their benefits in terms of reduced SEE sensitivity make them attractive; however, more studies are required to assess whether they are sufficiently robust against TID effects.

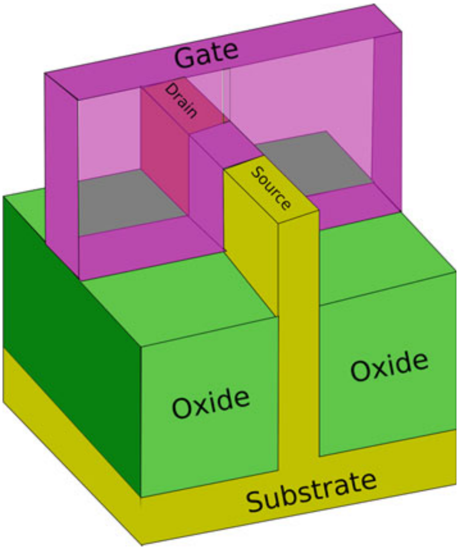
Currently, the move away from bulk planar transistors is perhaps the most effective mitigation against soft errors. In [107] the authors presented a concise overview of the SER benefits of different technologies including FDSOI and provide test data for nodes up to 65 nm. In Fig. 17, reproduced from [107], the relative SER benefit of multiple technologies is compared. In the following two sections, our goal is to present the latest results in SER analysis and measurement of soft errors in the FinFET and FDSOI technologies, respectively.

Another trend in terrestrial applications is that circuits in advanced technologies are being increasingly used in safety critical applications such as automotive and

Fig. 17 SER overview of multiple technologies up to 65 nm (reproduced from [107])

Process Option	Relative SER
Bulk General Purpose	100
Bulk Low Power	90
Triple/Deep N-Well	60-75
Body tied PD SOI	<1
Floating body PD SOI	15-20
Double Gate	2
Addition of MIM caps (eDRAM process)	0 (alpha) <1-10 (neutron)

Fig. 18 Overview of FinFET device



industrial automation. It is currently estimated that over half of the end points in the Internet of Things (IoT) will be safety critical, thus a careful understanding of the impact of radiation effects is required in order to assess their impact on reliability and safety goals. Despite the SER benefits achieved at the process technology level, there is still a need for circuit-level techniques. It is a common practice to protect memories using error correcting codes (ECC) so the real challenge remains the protection of flip-flops and to a lesser extent combinatorial logic. In a subsequent section we present recent results in the design and test of hardened flip-flops for both traditional bulk technologies as well as FinFET and FDSOI technologies.

4.2 *Impact of FinFETs on SER*

The key characteristic of FinFET devices is that there is a “fin” which wraps around the conducting channel between the source and drain as shown in Fig. 18. The fact that the gate structure wraps around the channel reduces the leakage current and reduces short-channel effects.

Several studies have shown that the critical charge for FinFET devices is either similar [108] or slightly lower [109] than similar bulk devices. It has also been shown that the doping profiles of bulk and FinFET devices are relatively similar [109]. The differences in SER sensitivity are explained by the differences in charge collection because of the thin drain region and narrow connection to the substrate. The initial charge collection, which is dominated by drift, is not so different between planar and FinFET devices. However, in the FinFET, there is very little charge collection due to diffusion from the substrate [108].

One of the first studies of FinFET devices was by Intel [110]. Note that Intel refers to their FinFET devices as tri-gate devices. In this study, they report that the neutron SER of 22-nm tri-gate 6T SRAM cells is 3.5× lower than a planar 32 nm cell. The improvement in neutron SER of 22-nm tri-gate flip-flops was less, in the range of 1.5× to 4×. However, the tri-gate devices are shown to be 10× to 300× less sensitive to alpha SER. This study showed that MCU rates and the extent of MCUs are not significantly lower than in bulk devices.

In [111], Intel reports new test results for their 14-nm tri-gate devices which have taller and narrower fins and thus reduced charge collection. In this study, the neutron SER of the 14 nm devices is shown to be about one-eighth that of the 22 nm devices while the alpha SER was reduced by about 4×. In the accelerated testing, the extent of the MCUs in the 14 nm technology was similar to the 22 nm technology. Interestingly, during real-time testing, the 14 nm devices showed several MCU events with very large extent (5 and even 14 bits), which was above the expectations from the accelerated testing and modeling.

In [112] Samsung reports the SER sensitivity of SRAM cells implemented in their 14 nm FinFET process and they report a 5–10× reduction in sensitivity for fast neutrons and alpha particles, as compared to 28 nm planar devices. Interestingly, they report a much smaller change in the sensitivity to thermal neutrons. In this study, single-fin and two-fin devices are studied and the latter are slightly more sensitive which was also confirmed by TFIT simulation [113].

In [114] the authors present a heavy-ion study of flip-flops implemented in 28 nm bulk planar, 20 nm bulk planar, and 16 nm bulk FinFET processes operating at 900 mV. In general, the 20 nm devices have a cross section about 50% lower than the 28 nm bulk devices. For lower LETs, the FinFET devices showed a cross section that is well over an order of magnitude lower than the planar devices. Above a LET of 20 meV cm^2/mg, there was very little difference in the sensitivity between the different devices. The drain region of the FinFET is much smaller and lower LET particles must strike directly in this region to cause an effect. At higher LETs, however, there is still significant charge collection in the substrate, thus the smaller difference in sensitivity. In space applications, low LET particles are dominant; however, the fact that at high LET there is less difference in sensitivity may reduce the SEE benefit of FinFETs in space applications.

The authors of [114] also performed TCAD simulations, building 3-D models using data from the PDK as well as predictive technology libraries. In these simulations, the ion track was simulated as a cylinder with the charge carriers following a Gaussian distribution. One of the key findings of this work was that the radius of the ion track plays a very important role in determining the sensitivity of FinFET devices. As the radius was swept from 5 to 50 nm, the impact on the SER of bulk devices was small; however, the diameter of the ion track radius played an important role for the FinFET devices. The simulation results highlight the difficulty in accurately simulating the effect of low LET ion strikes in FinFET devices.

In [115] the authors perform an in-depth study of SBU and MCUs for planar and 16 nm FinFET SRAMs. The test results show that between 20 nm planar and 16 nm FinFETs, there is an order of magnitude reduction in SBUs caused by alpha, thermal and fast neutrons. Furthermore, there is also an order of magnitude drop in the absolute rate of MCUs. In this work, it is also shown by TCAD simulations that MCUs in FinFETs are primarily due to charge sharing and that the increased doping levels that are used in FinFETs tend to reduce charge collection and lower the rate of MCUs.

The above works have primarily studied single-event effects on FinFET devices. In [116], the authors present a detailed study of the TID effect on FinFETs, particularly, the dependence on the number of fins, although the study is done for an older 90 nm technology. TID generally causes positive traps in the oxides and at the silicon to oxide boundaries. In this study, the authors find that the impact of TID on leakage current is greatest for single-fin devices. The single-fin devices show the largest increase in leakage current and the largest shift of V_t, compared to two- and 40-fin devices.

To summarize, it is clear that FinFET devices show a very significant reduction in SER compared to planar devices. The contribution of alpha SER is much lower than for planar devices. It also interesting to note that Intel also reports significant improvements in other reliability metrics such as TDDB, BTI, HCI, and SILC [117].

4.3 Impact of FDSOI on SER

An excellent overview of the SER benefits of FDSOI technology is presented in [107]. SOI has long been known to provide strong protection against radiation effects; however, it has generally been significantly more expensive than bulk technologies and used only in specialized applications. Recently, ST Microelectronic's 28 nm FDSOI technology, which is described in detail in [118], has brought this technology more into the mainstream. In this technology, because of the thin box, these devices have an ultra-thin body. The field between the source and drain is confined between the gate oxide and the box, making the transistor behavior closer to ideal. In terms of radiation effects, the sensitive volume is isolated from the substrate, making the sensitive area extremely small.

In [119] it is reported that the alpha particle SER sensitivity of ST's 28 nm FDSOI technology is approximately 1 FIT/Mbit, which is about two orders of magnitude lower than similar 28 nm bulk technologies, although at lower voltages (0.8 V), the alpha SER does increase (4×…8×) [120]. It is reported [120] that this technology has a raw neutron SER of approximately 10 FIT/Mbit, which is about

20× lower than comparable bulk technologies. The technology also has a low sensitivity to thermal neutrons (2 FIT/Mbit) [121]. A further benefit is that the technology is immune to SEL [120], even at high temperature, which is to be expected, as the parasitic thyristor structure does not exist. Taken together, these characteristics make this technology attractive for applications which require a low sensitivity to radiation effects. Investigations are underway to potentially qualify the technology for space applications; however, this requires a better understanding of the TID effects and also an investigation to better understand the SEE benefits in harsh radiative environments.

4.4 SOI FinFETs

IBM is developing processes to build advanced FinFET transistors in an SOI process. In [122], detailed simulation results of the sensitivity of these devices are presented, and as might be expected, they show extremely low radiation sensitivity. In this paper, it is predicted that the PDSOI FinFET SRAM cells will be two orders of magnitude-less sensitivity than planar PDSOI cells, which already have a very low sensitivity. The critical charge of these cells is expected to be approximately 4 fC, nearly an order of magnitude higher than the 22 nm planar devices.

Although SOI FinFET devices have extremely low SEE sensitivity, preliminary studies [123] show that they are sensitive to TID. The study in [123] analyzed the effect of TID on 14 nm SOI FinFETs, 14 nm bulk FinFETs, and 22 nm UTBB FETs with two different box thicknesses. Interestingly, the impact of TID was quite different across these devices. It was found that for the SOI FinFETs, a V_t shift of 14 mV was observed after 100 krad and these transistors were most sensitive to TID in the off state. For the bulk FinFETs, there was very little shift in V_t; however, the off-state current increased dramatically. The UTBB FETs showed a significant V_t shift with dose, with a sensitivity greater than the bulk FinFETs.

At this point, it is clear that both FDSOI and FinFET devices bring huge benefits in terms of SEE sensitivity. The TID analysis of these technologies in small geometries is still underway, but it does appear that they are quite sensitive which may be an obstacle for their adoption for space applications. For terrestrial applications, however, they provide a massive benefit due to their extremely low rate of soft errors.

4.5 Hardened Cells

For many terrestrial applications, such as networking or general-purpose computing, the large soft-error benefit provided by advanced process technologies is such that it may not be necessary to use hardened flip-flops in order to obtain reliability targets. On RAMs, the use of ECC remains a good practice as ECC has a relatively low cost and can correct errors from any source, whether it be radiation effects, RTN, aging, or other faults. Furthermore, in today's SoCs, RAMs represent the majority of the die area, and thus this simple technique can provide a high overall level of protection.

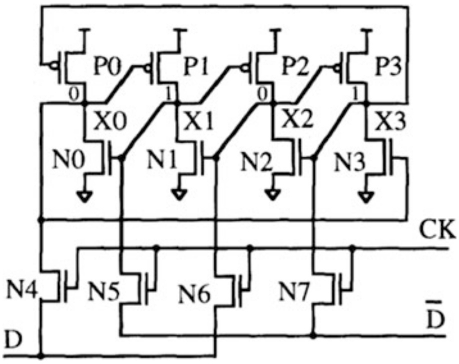
For high-reliability applications, such as automotive, even when advanced process technologies with low soft-error sensitivity are adopted, there is still a need for hardened flip-flops to protect the most functionally critical state in the logic. This is partly due to the fact that the number of flip-flops per chip increases with scaling and it is typical to have SoCs with tens of millions of flip-flops. This is also the result of new safety standards, such as ISO26262, which require a systematic analysis of the effects of faults.

The most widely used techniques for hardening flip-flops include DICE [124], LEAP [125], increased nodal capacitance, Quatro [126, 127], reinforcing charge collection (RCC) [128], device stacking [129, 130], guard gates [131], variants on DICE [132], or TMR designs.

The classic DICE flip-flop is illustrated in Fig. 19 and, as is well known, provides immunity against upsets to a single node. In older technologies, the DICE design could provide a reduction up to 1000× in SER sensitivity. However, recent studies have shown [133] that even at 28 nm the benefit of the DICE is limited. In advanced technologies, a single particle can deposit charge on multiple nodes and, due to this charge sharing, in order to achieve the benefit of DICE, the layout must be carefully optimized using techniques such as LEAP [125]. With careful layout, it is still possible to design hardened flip-flops that can achieve two orders of magnitude in soft-error sensitivity; however, the benefits are less for high LET particles.

Particles that strike the device at normal incidence are much less likely to deposit charge on multiple nodes, whereas particles that strike at an angle often upset multiple nodes. When evaluating the sensitivity of hardened flip-flops, especially for space applications, it is important to analyze the effect of angular strikes on the design. In Fig. 20, the simulated effect of a heavy-ion strike is shown at normal incidence and at a tilt of 60°. The colors represent the sensitive cross section, and as can be clearly seen, the design is significantly more vulnerable to angular strikes.

Fig. 19 Schematic of DICE flip-flop



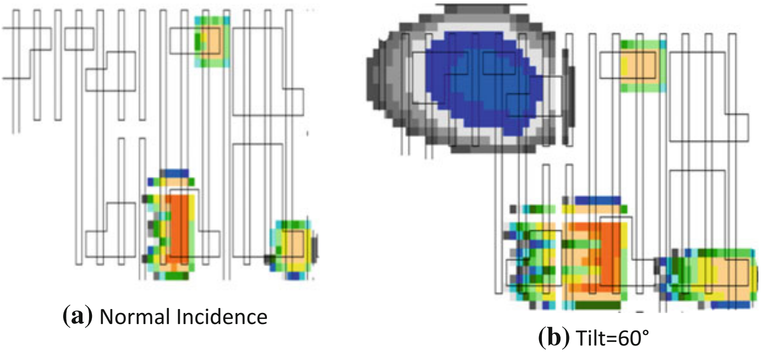


Fig. 20 Simulated heavy-ion strike on DICE FF using TFIT [113]

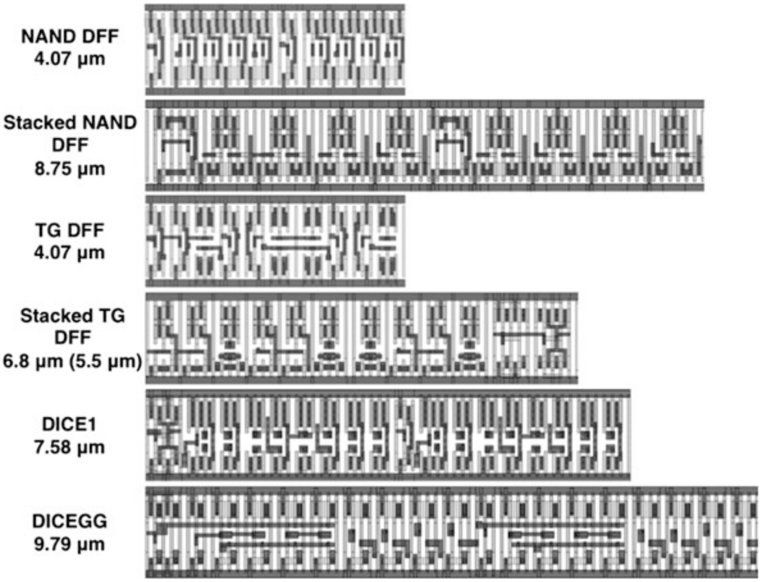


Fig. 21 Layout of six flip-flops in 32 nm FDSOI (reproduced from [129])

A recent test chip in a 32 nm FDSOI technology was implemented by the authors of [129]. The chip consisted of six different flip-flop designs. Two of the designs (NAND and transmission gate—TG) were unhardened. Two other designs were based on the DICE technique, one of which implemented guard gates [131]. Finally, an alternate implementation of the unhardened flip-flops was implemented using stacked transistors The layouts of the six designs are reproduced from [129] in Fig. 21. The large area overhead for hardened flip-flops is clearly visible (Fig. 22).

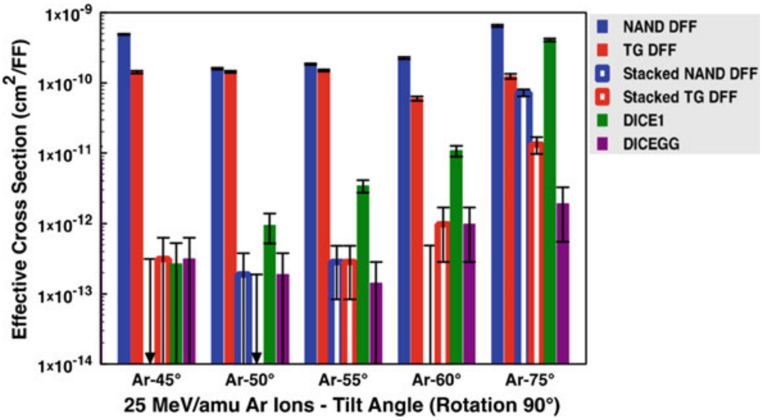


Fig. 22 Heavy ion test results of six 32 nm FDSOI flip-flops (reproduced from [129])

The test results showed that the DICE designs were not completely immune to alpha particles, although their sensitivity was reduced by over two orders of magnitude. When tested under heavy ions, the DICE designs showed increasing sensitivity with angular strikes, as was observed in simulation results shown earlier (for a different technology). Overall, in this study, the stacked transistor design performed better than the DICE designs, especially for particles arriving at a high tilt.

Of course, all hardened flip-flops induce penalties in area, power, and timing. In [134], the authors present a broad study of 30 different industrial flip-flops, including 11 hardened designs, implemented in a 28 nm bulk process. In this study, it is reported that the average area overhead is 3.8×, the average power overhead is 2.5× and the average timing (CLK → Q) is 1.2×. Given the high cost of hardened flip-flops, it is important to carefully select the most functionally critical flip-flops in the design, which requires analysis techniques [135].

At this point, the reader will appreciate that there are a large number of techniques available for designing flip-flops with reduced SER. It is beyond of the scope of this book to provide a comprehensive review of all techniques; however, the reader can find more information in the referenced works. The “best” cell design for a given application depends on many factors including the acceptable area and power penalties, the radiative environment, and the required level of protection. Simulation and testing are essential when designing and validating radiation-hardened flip-flops.

In an unprotected logic design (excluding RAMs), the largest overall contribution to soft errors comes from flip-flops. Using hardened cells, the contribution of flip-flops to the overall error rate can be managed. Careful selection of which flip-flops to harden can keep the area penalties reasonable. Although the focus is often on the actual storage cell, as shown in [136], the design of the clock tree plays an important role in reducing the rate of upsets in flip-flops. Also, after flip-flops have been protected, the relative contribution of combinatorial logic gates increases and designers must pay attention to SETs.

5 Mitigation of Voltage Droop

Traditionally, the voltage droop phenomenon has been an important reliability factor in the power delivery subsystem of chips and has been mitigated by off-chip schemes and on the board itself. However, with the technology scaling to nanoscale dimensions and the increase in transistor density per die along with the increase of chip frequency, the off-chip techniques have become not enough alone; and advanced mitigation techniques have also emerged inside the chip. This section covers the main techniques to either avoid or mitigate voltage droop in modern electronic chips.

5.1 Classification

Mitigation schemes for voltage droop in modern integrated circuits can be classified into two categories:

- Off-Chip techniques: These methods aim at improving the supply voltage network impedance and reducing the voltage variation on the board power delivery subsystem. They are generally utilized to avoid low and medium frequency voltage droops.
- On-Chip techniques: These are techniques applied inside the chip to reduce the supply voltage droops within the die and mitigate their effects. They have obtained significant importance due to the increase in different variation sources and complexity in modern chips. Note that on-chip methods are generally applied to avoid high-frequency voltage droops.

Figure 23 outlines the main off-chip (on the board) and on-chip (inside the die itself) voltage droop mitigation techniques. Next, these schemes are discussed and the focus will be on the on-chip voltage droop compensation approaches, as they are more efficient in terms of power and performance inside the modern chips.

5.2 Off-Chip Techniques

The most important factor in terms of voltage droop for a chip on the board is the voltage on its pads. If there were no current flow in the power delivery network

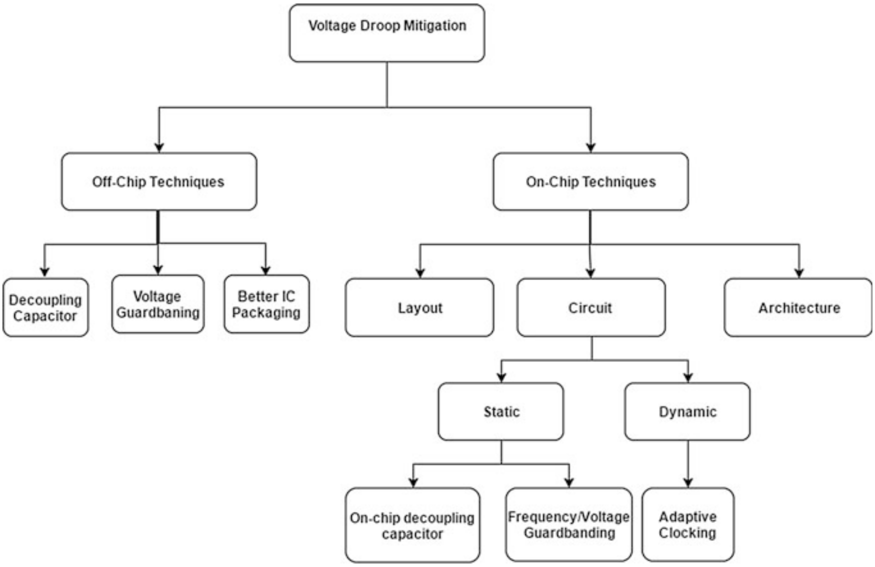
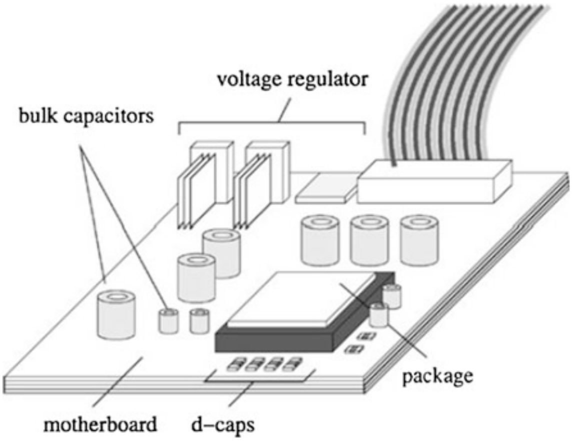


Fig. 23 Overview of existing voltage droop mitigation techniques

Fig. 24 Microprocessor power delivery subsystem



interconnects, the voltage would appear constant on the chip pads. Figure 24 exhibits an example of the microprocessor power delivery subsystem and the specific components in relation with the processor power delivery [137].

Any improvement in the components of power delivery on the board can lead to reduction of voltage droops. For instance, enhanced voltage regulator modules can better mitigate the low-frequency voltage droops [137]. In this sense, the off-chip techniques which can be utilized to reduce the voltage droops are discussed next.

Decoupling Capacitors: Adding decoupling capacitors (d-caps in Fig. 24) can reduce the power supply impedance and make the load less sensitive to existing inductance in the power pads. The off-chip decoupling capacitors are very efficient in avoiding the on-board voltage droops at mid-range frequencies. Moreover, the effectiveness of decoupling capacitors at high frequency is greatly increased when the inductance in the power delivery path is minimized [138].

Voltage Guardbanding: This approach can be classified into two methods. In the first one, also called static voltage margining, a voltage higher than the nominal one is set on the board by the voltage regulator module. However, in the second technique, the voltage on the board is increased by the regulator during the periods in which the processor has a low activity, so that the potential sudden voltage droops can decrease. Note that the voltage regulators typically have slow response frequencies and cannot compensate high-frequency V_{dd} droops [139]. The on-board voltage guardbanding methods impose additional power loss specifically during the chip low loads.

Better IC Packaging: As the quality of the chip packaging improves, the parasitic effects in the chip interconnects become less, which reduces the voltage droops. Moreover, the vias are placed close together to minimize the inductance effect [139]. These improvements significantly reduce the potential occurrence of the voltage droops.

5.3 On-Chip Techniques

The concern of the on-chip power supply droops has increased with the shrink of the CMOS feature size and increase of the frequency. This has led to novel on-chip techniques to mitigate its effect, which are classified into layout-, circuit-, and architecture-based solutions described in the following sections.

Layout-Based Solutions: Without a careful layout planning, the design may suffer from power supply noise and potential supply voltage droops [140]. By considering the power supply planning during the early design stage inside the chip the circuit block locations and shapes can be flexibly changed to minimize the droop phenomenon. For instance, noticing the distance of the power pads from the potential high switching activity nodes in the chip and keeping these pads close to each other can help in avoiding the voltage droops. Therefore, having the power lines as close as possible to the chip blocks by utilizing multiple supply voltages and ground pins in the floor plan of the chip can help in reducing the dynamic variations inside the chip. Figure 25 shows an example of an advanced power supply distribution inside the chip utilizing the IBM floor-planning standard (C4). It depicts that how the power and signal pads can be distributed to reduce the potential voltage droops.

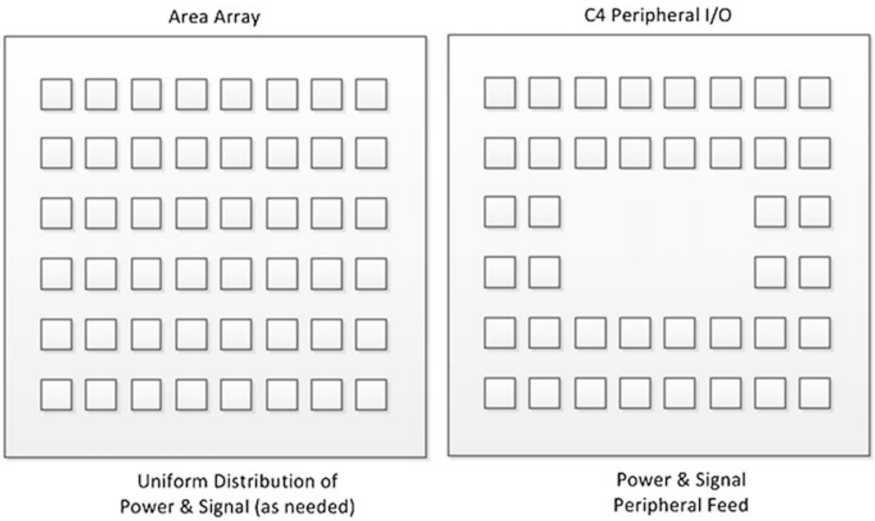


Fig. 25 Chip floor-planning showing power distribution patterns from IBM

5.4 Circuit-Based Solutions

With the technology scale reaching the nanoscale design era, circuit-based techniques have obtained significant importance; they are the ones which can significantly reduce the impacts of the voltage droop at high frequencies. Note that the circuit-based techniques can be categorized into two groups:

- Static (pre-silicon) techniques: Static circuit-based solutions are generally designed for the chip's worst-case operational conditions; therefore, they might be pessimistic and not efficient in terms of performance or energy consumption. Moreover, they require proper modeling of the power delivery network, which might be quite complex in modern chips.
- Dynamic (post-silicon) techniques: Dynamic circuit-based voltage droop mitigation techniques consider the chip's runtime operational conditions to apply the appropriate mitigation margin (reducing the frequency or increasing the supply voltage) respectively. They can adapt themselves to the on-chip supply voltage variations and compensate its effect for a robust operation.

  In the following, first the static techniques including on-chip decoupling capacitors and frequency or voltage guardbanding are described. Thereafter, the dynamic approach of adaptive clocking is discussed.

On-Chip Decoupling Capacitors: According to modern scaling trends, on-chip decoupling capacitors must be added inside the die to suppress the droops and reduce the noise in it [141]. They function based on providing charge to circuits upon a sudden current demand [142]. Figure 26 shows an on-die distributed grid model of the parasitics inside the chip, including the decoupling capacitors [137,

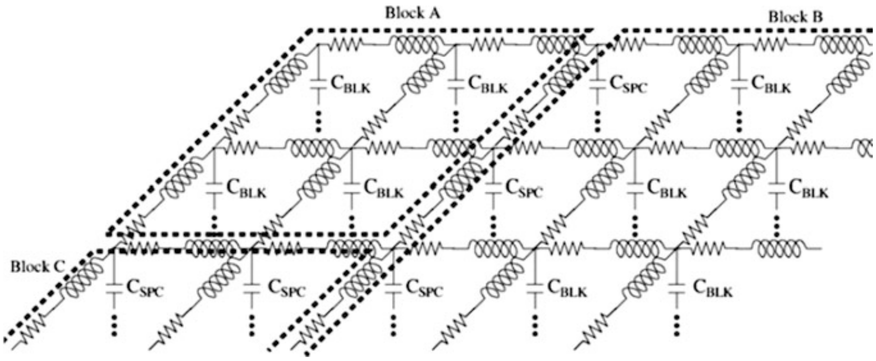


Fig. 26 On-die distributed grid model containing additional in-die decoupling capacitors

138]; C_{spc} represents the decoupling capacitance located between the functional units and C_{blk} represents the intrinsic parasitic capacitance of the functional units.

Although the on-chip decoupling capacitors can balance the abrupt changes in power delivery of chip blocks, their implementation results in more cost in terms of area and leakage when the chip size reduces. Moreover, these on-chip decoupling capacitors have some imperfections, which can lead to additional voltage resonances [143].

Frequency (F_{CLK})/Supply Voltage (V_{dd}) Guardbanding: To ensure reliable operation of the microprocessors in the existence of voltage droops, conventionally the design is built with guardbands in the operating clock frequency (F_{CLK}) or supply voltage (V_{dd}) [144]. This inflexible approach can limit the exploitation of the high-performance mode of the microprocessor by setting its operational frequency to the worst-case of supply voltage variation [145]. Furthermore, the inability to reduce the V_{dd} during favorable operating conditions decreases the energy efficiency of the chip. Note that these marginal F_{CLK} or V_{dd} guardbanding in modern microprocessors can lead to even higher guardbands than previous designs, therefore, making it necessary to design dynamic approaches, which can significantly reduce the guardbands.

Adaptive Clocking: This dynamic technique is the most important circuit-based approach to mitigate the voltage droops in modern microprocessors and has been utilized in various industrial products such as in AMD, Intel, and ARM microprocessors [146–148]. It is based on adjusting the clock period in relation with voltage variations, so that the clock runs at a lower frequency until the supply voltage returns to the nominal value [149]. The adaptive clocking technique can be categorized into two major classes:

- Traditional on-die monitor-based schemes: This technique relies on sensors to detect the droop and then adapts the frequency accordingly to mitigate the droop.
- Modern adaptive clock distribution-based schemes: This approach utilizes an in situ monitoring approach to reduce the delay between the droop detection and the frequency adaptation.

Both techniques are discussed in more detail in the following.

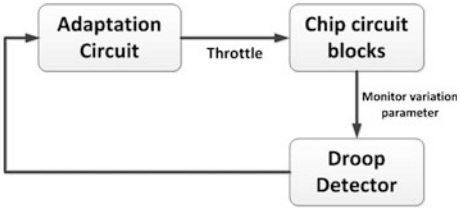
Traditional on-die monitors: The conventional dynamic approach is based on utilizing on-die sensors inside the chip, to measure specific parameters such as voltage or current or temperature [140, 141]. Then, these monitors are interfaced with adaptive control circuits to react to existing variations, by adjusting the operating parameters such as the F_{CLK} or V_{dd}. For instance, the frequency of the chip will be adapted with the droop in such a way that no processing error occurs. Figure 27 shows an example framework for this approach, where the droops inside the circuit blocks are detected by a detector, which will then stimulate the adaptation circuits to mitigate the impact of the voltage droop.

The conventional on-die sensors and adaptive circuits need sufficient time in order to respond to parameter variations. However, in the presence of high-frequency V_{dd} droops the on-chip sensors and feedback-based adaptive circuits are not able to respond to fast variations. Therefore, still some F_{CLK} or V_{dd} guardbanding is necessary to guarantee a reliable chip operation, imposing performance and energy overheads.

Modern adaptive clock distribution: The second adaptive approach to mitigate the impact of supply voltage droops (high-frequency V_{dd} droops) is based on an all-digital dynamically adaptive clock distribution [137]. This technique prolongs the clock-data delay compensation in critical paths during a V_{dd} droop, by exploiting a tunable-length delay prior to the global clock distribution. The adaptive clock distribution design contains three major circuit blocks: 1-On-die Dynamic Variation Monitor (DVM), 2-Tunable-Length Delay (TLD), and 3-Clock gating circuit. Figure 28 shows a block diagram example of Intel test chip, fabricated utilizing this technique and including the corresponding monitoring and adapting blocks [137].

The impact of dynamic parameter variations on critical path timing margin is measured by the DVMs. Once a voltage droop is detected by DVM, the TLD, which is located between the clock generator and the global clock distribution, proactively gates the clock for the duration of the V_{dd} droop. TLD extends the delay

Fig. 27 Feedback loop in sensor-based V_{dd} droop mitigation technique



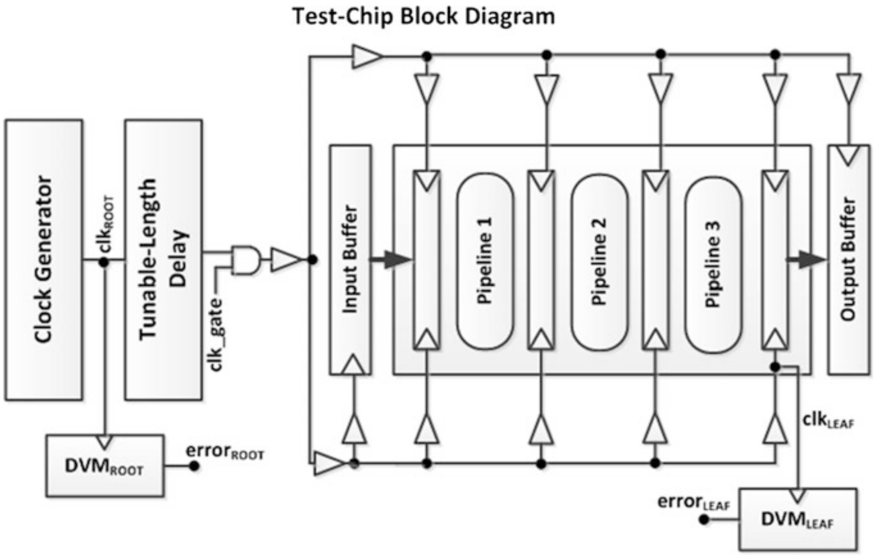


Fig. 28 Block diagram of the dynamically adaptive clock distribution technique

and changes the delay sensitivity to the V_{dd} in the clock distribution; therefore, it mitigates the impact of the V_{dd} droop. An alternative to the clock gating is to reduce the F_{CLK} in half with a clock divider circuit.

In comparison with other existing techniques, utilizing an adaptive clock distribution has significant advantages in terms of performance and energy efficiency by reducing the guardbands for potential V_{dd} droops. However, the main disadvantage of this approach is its need for a post-silicon calibration [145].

5.5 Architectural-Based Solutions

The architectural methods to mitigate the voltage droop in processors are generally known as resilient error detection and recovery approaches. They function based on two main concepts. The first is based on reducing the activity of the processor to avoid the droops, by throttling the instruction issues. Furthermore, the second approach allows the droops to occur inside the chip and then the processor has a built-in mechanism to recover its state and to correct the error [142].

As an example, [143] utilizes a resilient microarchitecture which can detect the induced timing violation by the dynamic variations. Then, it isolates the error from the corrupting architecture state and corrects the error through instruction replay. The error correction can occur during multiple cycles to prevent timing errors corrupting the architectural state of the processor.

The key advantage of the resilient error detection and recovery approaches is their ability to mitigate the guardbands for both fast and slow changing variations. Nevertheless, the main disadvantages are the design complexity overhead and the need for post-silicon calibration.

5.6 Summary

This section has covered the main techniques to either avoid or mitigate the supply voltage droops in modern microprocessors. The off-chip techniques have been traditionally used to reduce the supply voltage noise and deliver a clean voltage to the chip pads. However, with an increase of chip design complexity, frequency and number of transistors per die and the use of on-chip mitigating techniques have become inevitable. Among the on-chip approaches, adaptive clocking is the most significant and efficient method to mitigate the effects of voltage droops inside the chip and has been utilized in many modern microprocessors.

6 Conclusion

In this chapter, we have provided an overview of how some of the major challenges to IC reliability can be mitigated. In advanced processes, variability is becoming a key challenge and the chapter opened with a discussion of techniques to manage the impact of static and dynamic variability.

The problem of variability is compounded by aging effects and the evolution of transistor parameters over the lifetime of the device. The second section of the chapter discussed the challenges of transistor aging in-depth, including how effects such as BTI, HCI, RTN, and self-heating can be managed at the process level.

Advanced technologies such as FinFETs and FDSOI have a reduced sensitivity to radiation effects; however, they remain a real concern. These were discussed including how technology scaling is impacting the design of radiation-hardened cells. Finally, the chapter wraps up with a discussion of how the high power required for large SoCs can induce significant static and dynamic voltage drops, causing errors when the voltage at the transistors falls too low. Advanced techniques to manage both on- and off-chip voltage drop were discussed.

Taken together, it is clear that new process technologies are posing significant reliability challenges. This chapter has focussed on mitigation techniques at the process level and subsequent chapters will discuss mitigation techniques at higher levels in the design flow.

References

1. C. Auth et al., 45 nm High-k+ metal gate strain-enhanced transistors, in *2008 Symposium on VLSI Technology* (Honolulu, 2008), pp. 128–129
2. P.A. Stolk, F.P. Widdershoven, D.B.M. Klaassen, Device modeling of statistical dopant fluctuations in MOS transistors, in *1997 International Conference on Simulation of Semiconductor Processes and Devices, 1997. SISPAD '97* (Cambridge, 1997), pp. 153–156
3. M.D. Levenson, N.S. Viswanathan, R.A. Simpson, Improving resolution in photolithography with a phase-shifting mask. IEEE Trans. Electron Devices **29**(12), 1828–1836 (1982)
4. P. Yu, S.X. Shi, D.Z. Pan, Process variation aware OPC with variational lithography modeling, in *2006 43rd ACM/IEEE Design Automation Conference* (San Francisco, 2006), pp. 785–790
5. Y.H. Su, Y.C. Huang, L.C. Tsai, Y.W. Chang, S. Banerjee, Fast lithographic mask optimization considering process variation, in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (San Jose, 2014), pp. 230–237
6. A. Awad, A. Takahashi, S. Tanaka, C. Kodama, A fast process variation and pattern fidelity aware mask optimization algorithm, in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (San Jose, 2014), pp. 238–245
7. K. Yuan, J.S. Yang, D.Z. Pan, Double patterning layout decomposition for simultaneous conflict and stitch minimization. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **29** (2), 185–196 (2010)
8. K.P. Subramaniyan, P. Larsson-Edefors, Manufacturable nanometer designs using standard cells with regular layout, in *2013 14th International Symposium on Quality Electronic Design (ISQED)* (Santa Clara, 2013), pp. 398–405
9. M. Pons, F. Moll, A. Rubio, J. Abella, X. Vera, A. González, VCTA: a via-configurable transistor array regular fabric, in *2010 18th IEEE/IFIP International Conference on VLSI and System-on-Chip* (Madrid, 2010), pp. 335–340
10. D. Blaauw, K. Chopra, A. Srivastava, L. Scheffer, Statistical timing analysis: from basic principles to state of the art. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(4), 589–607 (2008)
11. M. Mani, M. Orshansky, A new statistical optimization algorithm for gate sizing, in *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004* (2004), pp. 272–277
12. J. Singh, V. Nookala, Z.-Q. Luo, S. Sapatnekar, Robust gate sizing by geometric programming, in *Proceedings of the 42nd Design Automation Conference, 2005* (2005), pp. 315–320
13. T. Burd, T. Pering, A. Stratakos, R. Brodersen, A dynamic voltage scaled microprocessor system, in *2000 IEEE International Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC* (San Francisco, 2000), pp. 294–295
14. M. Elgebaly, M. Sachdev, Variation-aware adaptive voltage scaling system. IEEE Trans. Very Large Scale Integr. VLSI Syst. **15**(5), 560–571 (2007)
15. M. Wirnshofer, L. Heiß, G. Georgakos, D. Schmitt-Landsiedel, A variation-aware adaptive voltage scaling technique based on in-situ delay monitoring, in *2011 IEEE 14th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)* (Cottbus, 2011), pp. 261–266
16. M. Ahuja, S. Narang, S. Patnaik, A process corner detection methodology for resilience towards process variations using adaptive body bias, in *2015 International Conference on Circuit, Power and Computing Technologies (ICCPCT)* (Nagercoil, 2015), pp. 1–6
17. J.W. Tschanz et al., Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. IEEE J. Solid-State Circuits **37**(11), 1396–1402 (2002)

18. S. Ghosh, R. Kaushik, Exploring high-speed low-power hybrid arithmetic units at scaled supply and adaptive clock-stretching, in *2008 Asia and South Pacific Design Automation Conference* (Seoul, 2008), pp. 635–640
19. S. Ghosh, S. Bhunia, K. Roy, CRISTA: a new paradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **26**(11), 1947–1956 (2007)
20. D. Ernst et al., Razor: a low-power pipeline based on circuit-level timing speculation, in *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36* (2003), pp. 7–18
21. M. Choudhury, V. Chandra, K. Mohanram, R. Aitken, TIMBER: Time borrowing and error relaying for online timing error resilience, in *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)* (Dresden, 2010), pp. 1554–1559
22. J.C. Smolens, B.T. Gold, B. Falsafi. J.C. Hoe, Reunion: Complexity-Effective Multicore Redundancy, in *2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)* (Orlando, 2006), pp. 223–234
23. N. Kandasamy, J.P. Hayes, B.T. Murray, Transparent recovery from intermittent faults in time-triggered distributed systems. IEEE Trans. Comput. **52**(2), 113–125 (2003)
24. B. Kaczer et al., The defect-centric perspective of device and circuit reliability—from gate oxide defects to circuits. Solid State Electron. **125**, 52–62 (2016)
25. G. Groeseneken et al., Achievements and challenges for the electrical performance of MOSFETs with high-k gate dielectrics, in *Proceedings of the International Conference on Physical and Failure Analysis of Integrated Circuits (IPFA 2004)* (2004), pp. 147–155
26. M. Jo et al., Improved high-k/metal gate lifetime via improved SILC understanding and mitigation, in *IEEE International Electron Devices Meeting (IEDM), Techical Digest* (2011), pp. 18.3.1–18.3.4
27. S. Ramey et al., Intrinsic transistor reliability improvements from 22 nm tri-gate technology, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2013), pp. 4C.5.1–4C.5.5
28. K.T. Lee et al., Frequency dependent TDDB behaviors and its reliability qualification in 32 nm high-k/metal gate CMOSFETs, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2011), pp. 2A.3.1–2A.3.5
29. A. Bezza et al., Physical understanding of low frequency degradation of NMOS TDDB in High-k metal gate stack-based technology. Implication on lifetime assessment, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2015), pp. 5A.5.1–5A.5.5
30. C.L. Chen et al., The physical mechanism investigation of AC TDDB behavior in advanced gate stack, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2014), pp. 5B.5.1–5B.5.5
31. B.P. Linder, D.J. Frank, J.H. Stathis, S.A. Cohen, Transistor-limited constant voltage stress of gate dielectrics, in *Proceedings of the Symposium on VLSI Technology* (2001), pp. 93–94
32. B. Kaczer, A. De Keersgieter, S. Mahmood, R. Degraeve, G. Groeseneken, Impact of gate-oxide breakdown of varying hardness on narrow and wide nFET's, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2004), pp. 79–83
33. B. Kaczer et al., Impact of MOSFET oxide breakdown on digital circuit operation and reliability, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2000), pp. 553–556
34. B. Kaczer, R. Degraeve, E. Augendre, M. Jurczak, G. Groeseneken, Experimental verification of SRAM cell functionality after hard and soft gate oxide breakdowns, in *Conference on European Solid-State Device Research (ESSDERC)* (2003), pp. 75–78
35. J. Sune, E.Y. Wu, W.L. Lai, Successive oxide breakdown statistics: correlation effects, reliability methodologies, and their limits. IEEE Trans. Electron Devices **51**(10), 1584–1592 (2004)

36. S. Sahhaf et al., TDDB reliability prediction based on the statistical analysis of hard breakdown including multiple soft breakdown and wear-out, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2007), pp. 501–504
37. M.A. Alam, R.K. Smith, B.E. Weir, P.J. Silverman, Statistically independent soft breakdowns redefine oxide reliability specifications, in *International Electron Devices Meeting (IEDM) Technical Digest* (2002), pp. 151–154
38. J.H. Stathis, S. Zafar, The negative bias temperature instability in MOS devices: a review. Microelectron. Reliab. **46**(2–4), 270–286 (2006)
39. Y. Mitani, Influence of nitrogen in ultra-thin SiON on negative bias temperature instability under AC stress, in *International Electron Devices Meeting (IEDM) Technical Digest* (2004), pp. 117–120
40. B.P. Linder et al., Process optimizations for NBTI/PBTI for future replacement metal gate technologies, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2016), pp. 4B.1.1–4B.1.5
41. J. Franco et al., NBTI in Replacement Metal Gate SiGe Core FinFETs: Impact of Ge concentration, fin width, fin rotation and interface passivation by high pressure anneals, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2016), pp. 4B.2.1–4B.2.7
42. J.P. Colinge et al., Nanowire transistors without junctions. Nat. Nanotechnol. **5**, 225–229 (2010)
43. A. Veloso et al., Gate-all-around NWFETs vs. triple-gate FinFETs: junctionless vs. extensionless and conventional junction devices with controlled EWF modulation for multi-V_T CMOS, in *Proceedings of the Symposium on VLSI Technology* (2015), pp. T138–T139
44. M. Toledano-Luque et al., Superior reliability of junctionless pFinFETs by reduced oxide electric field. IEEE Electron Device Lett. **35**(12), 1179–1181 (2014)
45. B. Kaczer et al., Maximizing reliable performance of advanced CMOS circuits—a case study, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2014), pp. 2D.4.1–2D.4.6
46. D.P. Ioannou et al., A robust reliability methodology for accurately predicting Bias Temperature Instability induced circuit performance degradation in HKMG CMOS, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2011), pp. CR.1.1–CR.1.4
47. K. Zhao, J.H. Stathis, B.P. Linder, E. Cartier, A. Kerber, PBTI under dynamic stress: from a single defect point of view, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2011), pp. 4A.3.1–4A.3.9
48. S. Wang, D.S. Ang, G.A. Du, Effect of nitrogen on the frequency dependence of dynamic NBTI-induced threshold-voltage shift of the ultrathin oxynitride gate P-MOSFET. IEEE Electron Device Lett. **29**(5), 483–486 (2008)
49. R. Fernandez et al., AC NBTI studied in the 1 Hz–2 GHz range on dedicated on-chip CMOS circuits, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2006), pp. 1–4
50. T. Nigam, Pulse-stress dependence of NBTI degradation and its impact on circuits. IEEE Trans. Device Mater. Reliab. **8**(1), 72–78 (2008)
51. T. Grasser, B. Kaczer, H. Reisinger, P.-J. Wagner, M. Toledano-Luque, On the frequency dependence of the bias temperature instability, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2012), pp. XT.8.1–XT.8.7
52. L. Heiß et al., New methodology for on-chip RF reliability assessment, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2016), pp. 4C.5.1–4C.5.7
53. W. Arfaoui et al., Energy-driven Hot-Carrier model in advanced nodes, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2014), pp. XT.12.1–XT.12.5
54. M. Koyanagi, H. Kaneko, S. Shimizu, Optimum design of n^+-n^- double-diffused drain MOSFET to reduce hot-carrier emission. IEEE Trans. Electron Devices **32**(3), 562–570 (1985)

55. C. Hu, S.C. Tam, F.-C. Hsu, P.-K. Ko, T.-Y. Chan, K.W. Terrill, Hot-electron-induced MOSFET degradation—model, monitor, and improvement. IEEE J. Solid-State Circuits **20** (1), 295–305 (1985)
56. M. Cho et al., On and off state hot carrier reliability in junctionless high-K MG gate-all-around nanowires, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2015), pp. 14.5.1–14.5.4
57. M. Cho, E. Bury, B. Kaczer, G. Groeseneken, Channel hot carrier degradation and self-heating effects in FinFETs, in *Hot Carrier Degradation in Semiconductor Devices*, ed. by T. Grasser (Springer, 2014), pp 287–307
58. S. Kim, J. Lee, Hot carrier-induced degradation in bulk FinFETs. IEEE Electron Device Lett. **26**(8), 566–568 (2005)
59. Y.-K. Choi, D. Ha, E. Snow, J. Bokor, T.-J. King, Reliability study of CMOS FinFETs, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2003), pp. 7.6.1–7.6.4
60. D. Lee, S. Lee, C. Yu, J. Park, A guideline for the optimum fin width considering hot-carrier and NBTI degradation in MuGFETs. IEEE Electron Device Lett. **32**(9), 1176–1178 (2011)
61. W. Liu, K. Etessam-Yazdani, R. Hussin, M. Asheghi, Modeling and data for thermal conductivity of ultrathin single-crystal SOI layers at high temperature, IEEE Trans. Electron Devices **53**(8), 1868–1876 (2006)
62. S. Tyaginov et al., Understanding and modeling the temperature behavior of hot-carrier degradation in SiON nMOSFETs. IEEE Electron Device Lett. **37**(1), 84–87 (2016)
63. C. Prasad et al., Self-heat reliability considerations on Intel's 22 nm Tri-Gate technology, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2013), pp. 5D.1.1–5D.1.5
64. E. Bury et al., Characterization of self-heating in high-mobility Ge FinFET pMOS devices, in *Proceedings of the Symposium on VLSI Technology* (2015), pp. T60–T61
65. T. Takahashi, T. Matsuki, T. Shinada, Y. Inoue, K. Uchida, Comparison of self-heating effect (SHE) in short-channel bulk and ultra-thin BOX SOI MOSFETs: impacts of doped well, ambient temperature, and SOI/BOX thicknesses on SHE, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2013), pp. 7.4.1–7.4.4
66. K.O. Jeppson, C.M. Svensson, Negative bias stress of MOS devices at high electric fields and degradation of MNOS devices. J. Appl. Phys. **48**(5), 2004–2014 (1977)
67. S. Tyaginov et al., A predictive physical model for hot-carrier degradation in ultra-scaled MOSFETs, in *International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)* (2014), pp. 89–92
68. S. Maeda et al., Negative bias temperature instability in triple gate transistors, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2004), pp. 8–12
69. A.N. Tallarico et al., Impact of the substrate orientation on CHC reliability in n-FinFETs—separation of the various contributions. IEEE Trans. Device Mater. Reliab. **14**(1), 52–56 (2014)
70. B.N.J. Persson, Ph Avouris, Local bond breaking via STM-induced excitations: the role of temperature. Surf. Sci. **390**, 45–54 (1997)
71. J.W. Lyding et al., Ultrahigh vacuum–scanning tunneling microscopy nanofabrication and hydrogen/deuterium desorption from silicon surfaces: implications for complementary metal oxide semiconductor technology. Appl. Surf. Sci. **130–132**, 221–230 (1998)
72. K. Hess, I.C. Kizilyalli, J.W. Lyding, Giant isotope effect in hot electron degradation of metal oxide silicon devices. IEEE Trans. Electron Devices **45**(2), 406–416 (1998)
73. K. Seo, R. Sreenivasan, P.C. Mclntyre, K.C. Saraswat, Improvement in high-k (HfO_2/SiO_2) reliability by incorporation of fluorine, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2005), p. 420
74. H.-H. Tseng et al., Defect passivation with fluorine in a Ta_xC high-K gate stack for enhanced device threshold voltage stability and performance, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2005), pp. 696–699

75. C.G. Van de Walle, W.B. Jackson, Comment on ‘Reduction of hot electron degradation in metal oxide semiconductor transistors by deuterium processing’, Appl. Phys. Lett. 68, 2526 (1996)]. Appl. Phys. Lett. **69**, 2441 (1996)
76. I.C. Kizilyalli, J.W. Lyding, K. Hess, Deuterium post-metal annealing of MOSFET’s for improved hot carrier reliability. IEEE Electron Device Lett. **18**(3), 81–83 (1997)
77. K. Onishi et al., Bias-temperature instabilities of polysilicon gate HfO_2 MOSFETs. IEEE Trans. Electron Devices **50**(6), 1517–1524 (2003)
78. N. Kasai, P.J. Wright, K.C. Saraswat, Hot-carrier-degradation characteristics for fluorine-incorporated nMOSFET’s. IEEE Trans. Electron Devices **37**(6), 1426–1431 (1990)
79. A. Shickova et al., Novel, effective and cost-efficient method of introducing fluorine into metal/Hf-based gate stack in MuGFET and planar SOI devices with significant BTI improvement, in *Proceedings of the IEEE Symposium on VLSI Technology* (2007), pp. 112–113
80. A. Veloso et al., Thermal and plasma treatments for improve d (sub-)1 nm EOT planar and FinFET-based RMG high-k latest devices and enabling a simplified scalable CMOS integration scheme, in *International Conference on Solid State Devices Materials (SSDM)* (2013), pp. 590–591
81. T. Grasser et al., Gate-sided hydrogen release as the origin of “permanent” NBTI degradation: from single defects to lifetimes, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2015), pp. 20.1.1–20.1.4
82. T. Aichinger, S. Puchner, M. Nelhiebel, T. Grasser, H. Hutter, Impact of hydrogen on recoverable and permanent damage following negative bias temperature stress, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2010), pp. 1063–1068
83. T. Grasser et al., The paradigm shift in understanding the bias temperature instability: from reaction-diffusion to switching oxide traps. IEEE Trans. Electron Devices **58**(11), 3652–3666 (2011)
84. J. Franco et al., Understanding the suppressed charge trapping in relaxed- and strained-Ge/SiO_2/HfO_2 pMOSFETs and implications for the screening of alternative high-mobility substrate/dielectric CMOS gate stacks, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2013), pp. 15.2.1–15.2.4
85. B. Kaczer, A. Veloso, M. Aoulaiche, G. Groeseneken, Significant reduction of Positive Bias Temperature Instability in high-k/metal-gate nFETs by incorporation of rare earth metals. Microelectron. Eng. **86**(7–9), 1894–1896 (2009)
86. J. Franco et al., 6Å EOT $Si_{0.45}Ge_{0.55}$ pMOSFET with optimized reliability (V_{DD} = 1V): meeting the NBTI lifetime target at ultra-thin EOT, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2010), pp. 4.1.1–4.1.4
87. K. Xiong, J. Robertson, Passivation of oxygen vacancy states in HfO_2 by nitrogen. J. Appl. Phys. **99**(4), 044105 (2006)
88. D. Liu, J. Robertson, Passivation of oxygen vacancy states and suppression of Fermi pinning in HfO_2 by La addition. Appl. Phys. Lett. **94**, 042904 (2009)
89. S. Sahhaf et al., Correlation between the V_{th} adjustment of nMOSFETs with HfSiO gate oxide and the energy profile of the bulk trap density. IEEE Electron Device Lett. **31**(4), 272–274 (2010)
90. H. Arimura et al., Ge nFET with high electron mobility and superior PBTI reliability enabled by monolayer-Si surface passivation and La-induced interface dipole formation, in *IEEE International Electron Devices Meeting (IEDM)* (2015), pp. 21.6.1–21.6.4
91. K.T. Lee, H. Kim, J. Park, J. Park, Gate stack process optimization for TDDB improvement in 28 nm high-k/metal gate nMOSFETs, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2012), pp. GD.2.1–GD.2.4
92. Y.-T. Chen et al., Effect of NH_3 plasma nitridation on hot-carrier instability and low-frequency noise in Gd-doped high-k dielectric nMOSFETs. IEEE Trans. Electron Devices **58**(3), 812–818 (2011)

93. R. Degraeve, G. Groeseneken, R. Bellens, M. Depas, H.E. Maes, A consistent model for the thickness dependence of intrinsic breakdown in ultra-thin oxides, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (1995), pp. 863–866
94. A. Kerber et al., Strong correlation between dielectric reliability and charge trapping in SiO_2/Al_2O_3 gate stacks with TiN electrodes, in *Proceedings of the Symposium on VLSI Technology* (2002), pp. 76–77
95. A. Shickova et al., Dielectric breakdown study of multi-gate devices, in *7th European Workshop Ultimate Integration of Silicon (ULIS)* (2006), pp. 141–144
96. K.K. Hung, P.K. Ko, C. Hu, Y.C. Cheng, Random telegraph noise of deep-submicrometer MOSFETs. IEEE Electron Device Lett. **11**(2), 90–92 (1990)
97. E. Bury et al. Study of (correlated) trap sites in SILC, BTI and RTN in SiON and HKMG devices, in *Proceedings of the International Symposium on the Physical and Failure Analysis Integrated Circuits (IPFA)* (2014), pp. 250–253
98. B. Kaczer, M. Toledano-Luque, J. Franco, P. Weckx, Statistical distribution of defect parameters, in *Bias Temperature Instability for Devices and Circuits*, ed. by T. Grasser (Springer, 2014)
99. C. Prasad et al., Bias temperature instability variation on SiON/Poly, HK/MG and trigate architectures, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2014), pp. 6A.5.1–6A.5.7
100. P. Weckx et al., Characterization of time-dependent variability using 32k transistor arrays in an advanced HK/MG technology, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2015), pp. 3B.1.1–3B.1.6
101. J. Franco et al., RTN and PBTI-induced time-dependent variability of replacement metal-gate high-k InGaAs FinFETs, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2014), pp. 20.2.1–20.2.4
102. M. Toledano-Luque et al., Degradation of time dependent variability due to interface state generation, in *Proceedings of the Symposium on VLSI Technology* (2013), pp. T190–T191
103. M. Toledano-Luque et al., Depth localization of positive charge trapped in silicon oxynitride field effect transistors after positive and negative gate bias temperature stress. Appl. Phys. Lett. **98**, 183506 (2011)
104. J. Franco et al., SiGe channel technology: superior reliability toward ultra-thin eot devices—Part II: Time-dependent variability in nanoscaled devices and other reliability issues. IEEE Trans. Electron Devices **60**(1), 405–412 (2013)
105. C. Liu, K.T. Lee, S. Pae, J. Park, New observations on hot carrier induced dynamic variation in nano-scaled SiON/poly, HK/MG and FinFET devices based on on-the-fly HCI technique: the role of single trap induced degradation, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest* (2014), p. 34.6.1
106. B. Kaczer et al., Origins and Implications of Increased Channel Hot Carrier Variability in nFinFETs, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (2015), pp. 3B.5.1–3B.5.6
107. P. Roche, G. Gasiot, Impacts of front-end and middle-end process modifications on terrestrial soft error rate. IEEE Trans. Device Mater. Reliab. **5**(3), 382–396 (2005)
108. Y.P. Fang, A.S. Oates, Neutron-induced charge collection simulation of bulk FinFET SRAMs compared with conventional planar SRAMs. IEEE Trans. Device Mater. Reliab. **11** (4), 551–554 (2011)
109. J. Noh et al., Study of neutron soft error rate (SER) sensitivity: investigation of upset mechanisms by comparative simulation of FinFET and planar MOSFET SRAMs. IEEE Trans. Nucl. Sci. **62**(4), 1642–1649 (2015)
110. N. Seifert et al., Soft error susceptibilities of 22 nm tri-gate devices. IEEE Trans. Nucl. Sci. **59**(6), 2666–2673 (2012)
111. N. Seifert et al., Soft error rate improvements in 14-nm technology featuring second-generation 3D tri-gate transistors. IEEE Trans. Nucl. Sci. **62**(6), 2570–2577 (2015)

112. S. Lee et al., Radiation-induced soft error rate analyses for 14 nm FinFET SRAM devices, in *2015 IEEE International Reliability Physics Symposium* (Monterey, 2015), pp. 4B.1.1–4B.1.4
113. H. Belhaddad, R. Perez, M. Nicolaidis, R. Gaillard, M. Derbey, F. Benistant, Circuit simulations of SEU and SET disruptions by means of an empirical model built thanks to a set of 3d mixed-mode device simulation responses, in *Proceedings of RADECS*, 27–29 Sep. (2006)
114. P. Nsengiyumva et al., A comparison of the SEU response of planar and FinFET D flip-flops at advanced technology nodes. IEEE Trans. Nucl. Sci. **63**(1), 266–272 (2016)
115. Y.P. Fang, A.S. Oates, Characterization of single bit and multiple cell soft error events in planar and FinFET SRAMs. IEEE Trans. Device Mater. Reliab. **16**(2), 132–137 (2016)
116. I. Chatterjee, E.X. Zhang, B.L. Bhuva, D.M. Fleetwood, Y.P. Fang, A. Oates, Length and fin number dependence of ionizing radiation-induced degradation in bulk FinFETs, in *Proceedings of the IEEE International Reliability Physics Symposium (IRPS)* (Anaheim, 2013), pp. SE.8.1–SE.8.6
117. S. Ramey et al., Intrinsic transistor reliability improvements from 22 nm tri-gate technology, in *2013 IEEE International Reliability Physics Symposium (IRPS)* (Anaheim, 2013), pp. 4C.5.1–4C.5.5
118. N. Planes et al., 28 nm FDSOI technology platform for high-speed low-voltage digital applications, in *2012 Symposium on VLSI Technology (VLSIT)* (Honolulu, 2012), pp. 133–134
119. V. Malherbe, G. Gasiot, D. Soussan, A. Patris, J.L. Autran, P. Roche, Alpha soft error rate of FDSOI 28 nm SRAMs: Experimental testing and simulation analysis, in *2015 IEEE International Reliability Physics Symposium* (Monterey, 2015), pp. SE.11.1–SE.11.6
120. G. Gasiot, D. Soussan, M. Glorieux, C. Bottoni, P. Roche, SER/SEL performances of SRAMs in UTBB FDSOI28 and comparisons with PDSOI and BULK counterparts, in *2014 IEEE International Reliability Physics Symposium* (Waikoloa, 2014), pp. SE.6.1–SE.6.5
121. G. Gasiot, D. Soussan, J.L. Autran, V. Malherbe, P. Roche, Muons and thermal neutrons SEU characterization of 28 nm UTBB FD-SOI and Bulk eSRAMs, in *2015 IEEE International Reliability Physics Symposium* (Monterey, 2015), pp. 2C.2.1–2C.2.5
122. P. Oldiges et al., SOI FinFET soft error upset susceptibility and analysis, in *2015 IEEE International Reliability Physics Symposium* (Monterey, 2015), pp. 4B.2.1–4B.2.4
123. H. Hughes et al., Total ionizing dose radiation effects on 14 nm FinFET and SOI UTBB technologies, in *2015 IEEE Radiation Effects Data Workshop (REDW)* (Boston, 2015), pp. 1–6
124. T. Calin, M. Nicolaidis, R. Velazco, Upset hardened memory design for submicron CMOS technology. IEEE Trans. Nucl. Sci. **43**(6), 2874–2878 (1996)
125. H.-H.K. Lee, K. Lilja, M. Bounasser, P. Relangi, I.R. Linscott, U.S. Inan, S. Mitra, LEAP: Layout design through error-aware transistor positioning for soft-error resilient sequential cell design
126. Q. Wu et al., Supply voltage dependence of heavy ion induced SEEs on 65 nm CMOS bulk SRAMs. IEEE Trans. Nucl. Sci. **62**(4), 1898–1904 (2015)
127. S.M. Jahinuzzaman, D.J. Rennie, M. Sachdev, A soft error tolerant 10T SRAM bit-cell with differential read capability. IEEE Trans. Nucl. Sci. **56**(6), 3768–3773 (2009)
128. N. Seifert et al., On the radiation-induced soft error performance of hardened sequential elements in advanced bulk CMOS technologies, in *2010 IEEE International Reliability Physics Symposium (IRPS)* (Anaheim, 2010), pp. 188–197
129. J.S. Kauppila et al., Utilizing device stacking for area efficient hardened SOI flip-flop designs, in *2014 IEEE International Reliability Physics Symposium* (Waikoloa, 2014), pp. SE.4.1–SE.4.7
130. R.C. Quinn et al., Heavy ion SEU test data for 32 nm SOI flip-flops, in *2015 IEEE Radiation Effects Data Workshop (REDW)* (Boston, 2015), pp. 1–5

131. A. Balasubramanian, B.L. Bhuva, J.D. Black, L.W. Massengill, RHBD techniques for mitigating effects of single-event hits using guard-gates. IEEE Trans. Nucl. Sci. **52**(6), 2531–2535 (2005)
132. H.B. Wang et al., An SEU-tolerant DICE latch design with feedback transistors. IEEE Trans. Nucl. Sci. **62**(2), 548–554 (2015)
133. K. Lilja et al., Single-event performance and layout optimization of flip-flops in a 28-nm bulk technology. IEEE Trans. Nucl. Sci. **60**(4), 2782–2788 (2013)
134. N. Gaspard et al., Soft error rate comparison of various hardened and non-hardened flip-flops at 28-nm node, in *2014 IEEE International Reliability Physics Symposium* (Waikoloa, 2014), pp. SE.5.1–SE.5.5
135. A. Evans, M. Nicolaidis, S.J. Wen, T. Asis, Clustering techniques and statistical fault injection for selective mitigation of SEUs in flip-flops, in *2013 14th International Symposium on Quality Electronic Design (ISQED)* (Santa Clara, 2013), pp. 727–732
136. H.B. Wang et al., Single-event transient sensitivity evaluation of clock networks at 28-nm CMOS technology. IEEE Trans. Nucl. Sci. **63**(1), 385–391 (2016)
137. S. Bhunia et al., *Low Power Variation-Tolerant Design in Nanometer Silicon* (Springer, 2011)
138. M.S. Gupta et al., Understanding voltage variations in chip multiprocessors using a distributed power delivery network, in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition* (2007), pp. 1–6
139. L.D. Smith et al., Power distribution system design methodology and capacitor selection for modern CMOS technology. IEEE Trans. Adv. Packag. **22**(3), 284–291 (2002)
140. H.-M. Chen et al., Simultaneous power supply planning and noise avoidance in floorplan design. IEEE Trans. Comput. **24**(4), 578–587 (2005)
141. K.L. Wong et al., Enhancing microprocessor immunity to power supply noise with clock-data compensation. IEEE J. Solid-State Circuits **41**(4), 749–758 (2006)
142. T.R. Arabi et al., Design and validation of the Pentium III and Pentium 4 processors power delivery, in *Proceedings of the VLSI Symposium* (2002), pp. 220–223
143. M. Holtz et al., On-die CMOS voltage droop detection and dynamic compensation, in *Proceedings of the Great Lakes Symposium on VLSI* (2008), pp. 35–40
144. K. Bowman et al., Circuit techniques for dynamic variation tolerance, in *Proceedings of the Design Automation Conference* (2009), pp. 4–7
145. K. Bowman et al., Adaptive and resilient circuits for dynamic variation tolerance. IEEE Des. Test **30**(6), 8–17 (2013)
146. K. Wilcox et al., Streamroller module and adaptive clocking system in 28 nm CMOS. IEEE J. Solid-State Circuits **50**(1), 24–34 (2014)
147. K. Bowman et al., A 22 nm all-digital dynamically adaptive clock distribution for supply voltage droop tolerance. IEEE J. Solid-State Circuits **48**(4), 907–916 (2013)
148. D. Bull et al., A power-efficient 32 bit ARM processor using timing-error detection and correction for transient-error tolerance and adaptation to PVT variation, in *Proceedings of the IEEE International Solid-State Circuits Conference* (2010), pp. 284–285
149. A. Grenat et al., Adaptive clocking system for improved power efficiency in a 28 nm x-86-64 microprocessor, in *Proceedings of the IEEE International Solid-State Circuits Conference* (2014), pp. 106–107
150. M.R.C.M. Berkelaar, J.A.G. Jess, Gate sizing in MOS digital circuits with linear programming, in *Proceedings of the European Design Automation Conference, 1990., EDAC* (Glasgow, 1990), pp. 217–221
151. H. Tennakoon, C. Sechen, Gate sizing using Lagrangian relaxation combined with a fast gradient-based pre-processing step, in *IEEE/ACM International Conference on Computer Aided Design, 2002. ICCAD 2002* (2002), pp. 395–402
152. A.K. Murugavel, N. Ranganathan, Gate sizing and buffer insertion using economic models for power optimization, in *Proceedings of the 17th International Conference on VLSI Design, 2004* (2004), pp. 195–200

153. P. Roche, J.L. Autran, G. Gasiot, D. Munteanu, Technology downscaling worsening radiation effects in bulk: SOI to the rescue, in *2013 IEEE International Electron Devices Meeting* (Washington, DC, 2013), pp. 31.1.1–31.1.4
154. J. Tschanz et al., Adaptive frequency biasing techniques for tolerance to dynamic temperature-voltage variations and aging, in *Proceedings of the IEEE International Solid-State Circuits Conference* (2007), pp. 292–294
155. J. Zhao et al., Thermal aware voltage droop compensation for multicore architectures, in *Proceedings of the Great Lakes Symposium on VLSI* (2010), pp. 335–340
156. M.S. Gupta et al., An event guided approach to reducing voltage noise in processors, in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition* (2009), pp. 160–165
157. K. Bowman et al., A 45 nm resilient microprocessor core for dynamic variation tolerance. IEEE J. Solid-State Circuits **46**(1), 194–208 (2010)

Dependability Solutions

Salvatore Pontarelli, Juan A. Maestro and Pedro Reviriego

Abstract This chapter presents an overview of existing dependability solutions for multicore processing systems. In the first section, the existing techniques to protect processor cores both at the hardware and software level are discussed. Then the protection of the different memories that are present in a multicore is reviewed in the second section. Finally, the protection of the interconnections is covered in the last section.

1 Solutions for Processor Cores

As all electronic systems, processors are vulnerable to most kind of errors. The inherent complexity of processors (combinational and sequential logic, state machines, register files, buses, etc.) makes them even more sensitive, producing a wider range of consequences. While in most digital circuits reliability presents a binary output (given a set of inputs, the outputs do or do not match the correct behavior), the scenarios associated to a processor are more varied. The outcome of an error happening in a processor architecture can be classified as follows [1]:

- Masked error: The error does not propagate, and both the output and the processor context end with correct values.
- Unaffected Output: The output of the execution is correct, but at least one of the registers in the processor context is different than expected. This hidden error usually manifests in future executions.

S. Pontarelli (✉)
Consorzio Nazionale Interuniversitario Per Le Telecomunicazioni (CNIT), Rome, Italy
e-mail: salvatore.pontarelli@uniroma2.it

J.A. Maestro · P. Reviriego
Universidad Antonio de Nebrija, Madrid, Spain

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_5

- Affected Output: The execution ends properly, but the value of the output differs from the expected result.
- Crash: There is an unexpected termination of the execution, because of an invalid operation, illegal memory access, etc.
- Hang: The execution does not end because it is trapped in an infinite loop or similar problems with the control flow.

Even in the ideal case in which the error seems to have been masked (correct output and correct processor context), the error may have corrupted the system memory, making the problem harder to detect, and leading to a situation called *silent data corruption*. This may happen, e.g., when an error affects a register whose value is stored in the memory. Later, the error in the register may be masked (e.g., being overwritten with a correct value), thus leading to an apparent correct output and processor context. However, the wrong value has been able to exit the processor and corrupt the memory system, possibly affecting other processes in the architecture.

This complexity associated to processors makes error detection and correction hard, usually leading to large area and/or performance overheads. In order to analyze the different solutions, these can be classified in three groups [2]:

- Solutions at the architectural level. These usually consist of adding modular redundancy (e.g., several cores executing the same processes simultaneously) and then relying on compare and rollback operations.
- Solutions at the micro-architectural level. A single copy of the architecture is used, but this has been modified at the register transfer level to make it more reliable (e.g., adding redundancy to the instruction pipe or protecting the register file).
- Solutions at the software level. In this case, the architecture is not modified, and error correction and detection takes place by examining the execution of the software programs.

1.1 Architectural Solutions

Architectural solutions rely on modular redundancy to correct and detect errors. No improvements are usually included at the register transfer level since redundancy happens at a top level. Due to this reason, hardware overhead tends to be large. Solutions in this category consist in both techniques to initially detect errors, and techniques to recover from the detected errors.

1.1.1 Detection

Most of the detections techniques rely on some kind of redundancy. In this way, if there are several core replicas working at the same time, any discrepancy among them can be considered as an error. This is the most straightforward solution, but it is also the most costly. To mitigate this, another option is to have a single core and add the minimal hardware to find out when an error is present in that core.

Talking about redundancy, the standard approach has been to use Triple Modular Redundancy (TMR). This is a common practice that may be feasible in the most sensitive parts of the register transfer level. However, using TMR in a massive way, usually leads to unfeasible cost and complexity. At a modular level, TMR would imply having three identical cores, running simultaneously (see Fig. 1). Then, a voter would compare the outcome of the three execution threads in order to find discrepancies. This leads to two questions: What does it mean that several cores run "simultaneously"? What do we have to consider the "outcome" of the execution?

First, simultaneous execution means "lockstep" operation. Lockstep implies that all the processors execute the same thread at the same time, and they keep an identical state in each cycle. This operation mode can be difficult to achieve, especially if all the processors are located in the same board. In this situation, the access to shared resources may produce structural hazards, which most of the times lead to a desynchronization of the cores. In this situation, processors are usually forced to periodically synchronize, thus incurring in performance degradation.

Second, the outcome should at least comprise all the values calculated by the program and the whole processor context. In the same way, the processor context should include all the registers in the architecture, both data registers and control ones (state flags, stack pointer, etc.). If the different processors run in lockstep mode, this information would be enough to detect errors, providing that all outputs and contexts are compared in each clock cycle. However, many times this information cannot be compared in each cycle, due to performance issues and because not all the internal registers may be easily accessible from outside the processor. In these situations, an error in the processor may reach the memory system, producing what is called *silent data corruption*. Examining the processor context a posteriori

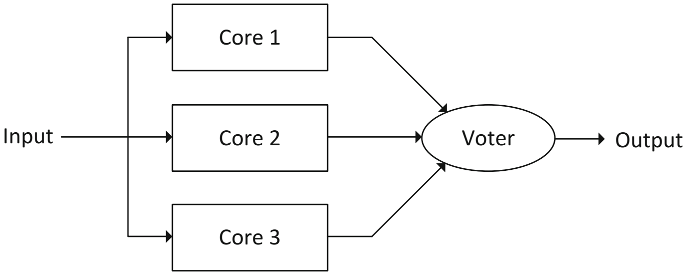


Fig. 1 Modular TMR

may not detect the error, since this may have been masked (e.g., by overwriting it with a correct value) after it has propagated to memory. In order to detect this kind of errors, the whole user memory space accessible by the program should also be checked. This increases the complexity of the process, and sometimes it may be unfeasible due to delay reasons.

As mentioned before, TMR is a straightforward mechanism to detect and correct errors, but most of the times it leads to unfeasible area overheads. To mitigate this, Dual Modular Redundancy (DMR) is sometimes the preferred option. In DMR, two identical cores are used, instead of three, thus reducing the area overhead. The previous considerations mentioned for TMR about lockstep operation and minimal information checking also applies. However, although the detection process is similar in TMR and DMR, correction is not trivial in the latter case.

There are alternatives to avoid the costly overhead presented by TMR and DMR.

Instead of using multiple processor replicas, redundancy can be achieved by running two identical threads in a single core. This technique is called *Redundant Multithreading* (*RMT*) [3, 4]. An example of this approach can be found in [5]. The proposal consists of creating a three-layer structure to support the system. The lower layer contains the standard architecture (processor, memory, and I/O). The upper layer is software based, and it is formed by two exact partitions that execute identical copies of the software application and a third partition that contains a checker whose mission is to compare the outputs of the other two. In the middle, there is another layer that is called the *hypervisor* layer. This layer provides services to support both the lower and upper layers. Respect to the lower one (standard architecture), it provides operating system services, as scheduling, resource management, etc., working in supervisor mode. Respect to the upper one (software), the hypervisor guarantees that the three partitions are mutually exclusive, with independent address spaces. The two program partitions send their outputs to the checker, using an interface provided by the hypervisor. If the checker determines that both coincide, the output is transferred to the lower layer. Otherwise, a signal error is flagged.

1.1.2 Correction

Once an error has been detected, it must be handled by the system. It is unclear at which level the error should be corrected, since that highly depends on the application itself. For example, when an error happens in a non-critical application, leading to an unexpected termination of the execution, the architecture level could ignore such an error. Then, the error handling process could be transferred to the Operating System, which would determine what to do in each situation. It could try to recover the error by itself (e.g., rescheduling the erroneous process for a second execution), or even passing the error to the user, who would determine how to proceed. However, most of the times this solution would not be feasible for a critical application, e.g., due to delay constraints or loss of unrecoverable data. In those situations, errors should be corrected at the architecture level [6], in the most transparent and fastest possible way. In the following, the latter approach (error handled within the architecture) will be the object of study.

From the previously explained detection techniques, TMR would present the most straightforward recovery mechanism. Since there are three exact replicas running an identical process, only a majority voter would be needed to trigger correction. While all the replicas produce identical outcomes (and, as explained before, the quantity of information would vary among cases), the system is supposed to be error-free. Once a discrepancy is detected, processors would vote through the majority voter. If only isolated errors are considered, then two of the outputs will be identical and the third one is different. The latter would be considered as the erroneous one, while the former would contain the right output and processor context. In this situation, the easiest way to correct the error would be to replace the wrong context of the processor that has failed with a copy of the correct context in the other cores. If the three processors are identical, saving and restoring the context in a different processor should be a simple task. In this way, the three processors can be put in lockstep again in a fast and transparent way.

Things are not so simple when less than three replicas are present in the system. If DMR is used, error detection is still simple, but there is no straightforward mechanism to determine which of the two incoherent contexts is the correct one and which is the wrong one. The simplest solution in this case would be to halt both processors, reset them, and start a brand new execution of the process. Unfortunately, this usually brings a performance degradation that is not acceptable in most cases. A better solution would be to "remember" intermediate states of the execution, so that the processors could return to one of them when an error is detected. This would also produce certain performance degradation, but that can be modulated by choosing the temporal distance of those intermediate states. This technique is called checkpointing, and it is a very used concept in architecture recovery mechanisms.

Checkpointing implies saving the whole processor context at regular intervals. Whenever an error is detected, the context associated to the most recent checkpoint can be restored, thus avoiding to start the execution from the beginning. The restoration of a previous stored context is called *rollback*. One important decision is to determine the *checkpoint interval*, i.e., the number of cycles between two consecutive checkpoints. If the interval is small, it may pose two much performance overhead, since context saving would consume too much time. On the other hand, if the interval is large, the processor will have to go back further when the context is restored, thus producing a performance degradation too.

Although this is an interesting approach to handle errors in the processor, checkpoints may also be corrupted by errors (see Fig. 2), making this technique inefficient under some circumstances. Let us define the *error window* as the amount of time that goes between the moment an error happens in the processor until it is detected. Even in the optimistic case in which the context of the different processors can be compared each clock cycle, there may be a lapse of time between error occurrence and detection. For example, an error that affects an instruction being

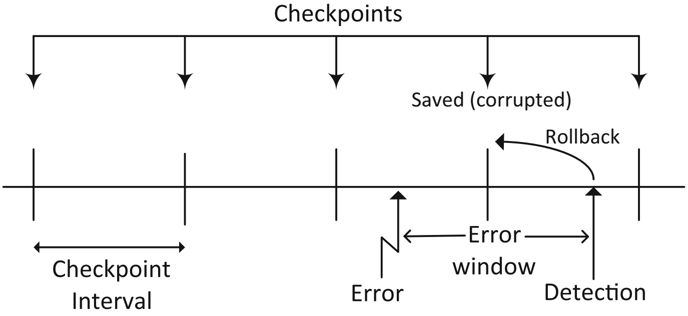


Fig. 2 Checkpoint corruption

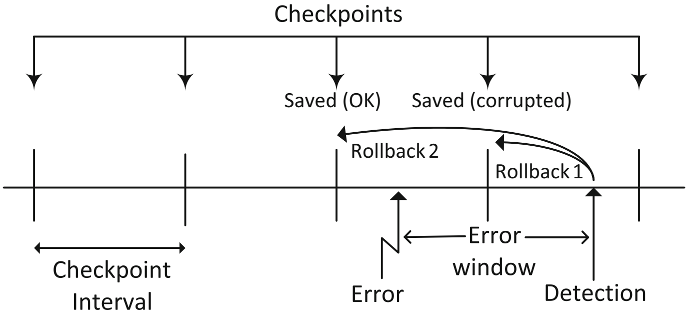


Fig. 3 Double checkpoint

fetched from memory may not manifest until several cycles later, when the instruction is in the last stages of the execution pipeline.

The relation between the checkpoint interval and the error window gives the probability of checkpoint corruption, in the following way. If an error happens in the processor and the next checkpoint save occurs in the subsequent error window, the stored context will be corrupted, since it contains the effect of the mentioned error. Then when the error window ends and the error is detected, the system would try to restore the last checkpoint context, but this would also contain that error, what would make the system fail again. This, if not properly handled, could induce the processor into an infinite loop.

A possibility to mitigate this effect is to keep the last two checkpoint contexts in the systems (see Fig. 3). In this way, when the processor detects an error and makes a rollback, if the error still persists that would mean that the last checkpoint was corrupted. Then the previous one would be restored in an attempt to take the processor to a valid state. This solution, of course, has a larger overhead than saving only the last checkpoint. Even with this, it is not guaranteed that the problem would disappear, since a given error window could potentially span more than two checkpoints intervals.

From the previous discussion, it is clear that determining the correct checkpoint interval is important to get the maximum reliability while keeping overheads low. Traditionally, fixed checkpoint intervals have been the common choice. This has proved to be the best option when failures arrive following a Poisson distribution [7]. However, recent studies suggest that a Weibull distribution would be a better option. In this case, it has been proved that dynamically scheduling checkpoints produces more benefits compared to the fixed checkpointing strategy [8]. This approach was implemented in ACR, an automatic checkpoint-restart system [9]. This system is resilient to both silent data corruption and crashes, and it is designed to automatically adjust the checkpoint interval depending on information from the environment. Also, depending on how this is done, different reliability levels can be achieved (strong, medium, and weak).

There are other approaches to error recovery that explore other techniques different than the standard TMR or DMR solutions. For example, the concept of *shadow replication*, that is indeed based on dual redundancy, but oriented to an energy-aware vision [10]. In this approach, each process has a so-called “shadow” process that runs in a different processor. Same thing as in the standard DMR, but in this case, this shadow process is executed at a slower speed. In this way, the power consumption of this process, and therefore of the system, is significantly decreased. In most of the cases, the original process will finish its execution without problems. When this happens, the associated shadow process is automatically terminated, thus reducing the overall energy consumption of the system. But in those cases in which the main process fails, the shadow process is promoted to main process, and continues the execution at a larger speed. Depending on the criticality of the task, several shadow processes may be linked to the main one establishing a hierarchy that would be recovered if consecutive errors affect the system. In this way, a customized approach can be followed in each case, tailoring reliability to the desired level.

Another interesting approach was presented in [11]. In this technique, two processors execute the same program, but in a different way. Therefore, this could be considered as a particular kind of diverse modular redundancy. The approach is based on DCE, a dual core execution system [12]. In this architecture, the front processor executes programs in a fast and speculative ways. After that, the process goes to the back processor that repeats the execution, using the information passed by the front processor, in a more detailed and un-speculative way. The architecture is designed in such a way to ensure rapid recovery from mis-speculations. However, discrepancies produced by error occurrences can be handled in the same way as the mentioned mis-speculations, thus achieving both fault tolerance and performance improvement with the same strategy.

1.2 Micro-Architectural Solutions: Processor Hardening

When we speak about micro-architectural solutions, we mean techniques that modify the register transfer level of the architecture, in order to provide error detection and correction capabilities that are not available in standard versions. To achieve this, two kinds of approaches may be followed. The first one is to add redundancy, for example TMR, but this time at a register transfer level instead of at a top level. Massive triplication is not usually an option, since the cost and complexity would be unfeasible. Therefore, only the most sensitive parts are selected for this triplication. Identification of these parts is not straightforward most of the times. This would require many experiments, usually utilizing a fault injector emulator, and several benchmarks and data loads to get precise results.

The second approach is to implement ad hoc solutions that change the architecture of the processor. This kind of approach is not as usual as the ones based on redundancy. Basically, the reason for this is that ad hoc protection techniques at the register transfer level are not as evident as adding some redundant cores or saving the context from one of them to the others. This kind of improvements implies understanding the architecture in detail, proposing modifications that can alter the basic behavior of the system. The thin trade-offs that usually exist in modern architectures make that any small change that is applied to a particular module may have negative side effects on the overall performance and area of the system.

There are several things that can be done in a processor in order to improve its fault tolerance. For example, one possibility is to protect data integrity while they are moved or stored within the architecture. Data standing in the register file or the cache memory are sensitive to errors, with a failure probability proportional to how long they are going to stay in those structures. Applying standard or modified Error Correction Codes (ECCs) to them is a common approach, as will be specified later in the chapter. Again, the overheads introduced by this approach may have a large impact on performance and area. Another possibility would be to set extra hardware to check certain functional units and operations and raise a flag in the case unexpected results are obtained. This is especially appropriate when processors run applications with some kind of algorithmic properties, as in this case there is a previous knowledge of how the behavior of the architecture should be. Deeper changes could imply modifying the number of stages and/or structure of the instruction pipe. In this way, longer pipes modified to carry out redundant and check operations can add reliability to the system, detecting situations that could have corrupted the instruction flow. Of course, this kind of decisions may have a large impact on the performance of the architecture, increasing data, and control hazards. Even the Instruction Set Architecture (ISA) can be modified to make it more fault tolerant. For example, when designing the microinstruction codification, micro-operations that are naturally more fault tolerant could be chosen. In this way, the traditional trade-off between performance and area made at the design time is now complicated with a new variable, the fault tolerance of the system. In the following, some of the techniques that handle micro-architectural reliability solutions will be presented. In order to make these techniques coherent, they are usually designed to handle not only processors but also the surrounding storing elements (e.g., cache memories). In this way, although these storing elements will be explained in depth in the next section, some details of particular architectures will be provided in the following for illustrative purposes.

Maybe, the most paradigmatic case of ad hoc protection in a processor is LEON [13]. This is a 32-bit processor, based on the SPARC-V8 RISC architecture, originally designed by the European Space Agency and Gaisler Research. Several releases of this processor exist, being LEON4 the most recent one. Among all these releases, LEON3FT is the one that has reported a higher fault tolerance. The architecture in this processor is an example of massive triplication at the register transfer level, since all flip-flops in the design are implemented with TMR. Apart from this, error detection and correction codes are exhaustively used in the architecture. Register file error correction is implemented, being able to handle up to 4 errors per 32-bit word. In the same way, the cache memory error correction is also available, with a limit of 4 errors per tag or 32-bit word. The error handling process is completely transparent, being all the operations performed internally to the architecture. These reliability mechanisms have been proved efficient against bit-flip errors. However, the system is still vulnerable to errors that produce latch-ups.

There are other cases of processors that have been designed with the principle of increasing fault tolerance. One example of this would be the HERMES processor [14]. In this processor, different approaches are combined inside the architecture in order to get a good reliability level. For example, the register file is protected with DMR, being both copies interleaved and using a parity bit. Also, there is a special port that facilitates background scrubbing operations. The speculative pipeline state is also protected using DMR. Both copies are compared at the end of each instruction, and only those with a positive match commit to architectural state. The rest of the registers outside the register file and the speculative pipeline are protected using TMR. The main memory is fully protected using error detection and correction codes. This allows a detection-only policy in the cache, since once an error is detected, the entry can be invalidated and recovered from main memory, assuming a write-through strategy. Finally, special instructions have been added to the architecture in order to facilitate repair operations.

Not all the approaches imply adding massive redundancy to the elements of a processor in order to increase reliability. Another possibility that requires minimal hardware overhead, and therefore improving the overall area and power consumption of the architecture, is called *dynamic verification*. This approach uses dedicated hardware to check, at runtime, the integrity of the so-called *system invariants*. These invariants have a known value when the processor is free of errors, and therefore they may be used to detect the occurrence of a failure. An example of this strategy may be found in DIVA [15]. In this architecture, the processor is complemented with the addition of a checker unit. This unit checks the computation of the main processor, and only those cases with a correct result are finally committed. The checker design is based on two check pipelines. The first pipeline verifies that the functional units that are producing calculations are working correctly. To do so, operations performed in the main processor are repeated in the checker just after the execution stage. In the case both results are different, an exception is triggered. A second pipeline verifies that the communication from the register file and memory with the main processor is correct, in order to check that the operands needed by the instructions have the right value. Again, any found discrepancy is signaled with an exception. Finally, in order to detect situations in which the main processor hangs or is trapped in an infinite loop, the checker operates a timer. The timer is set with the longest expected execution time for an instruction. If this time is exceeded, the checker assumes there is a problem with the control flow and launches an exception. In all the cases, it is assumed that the checker is implemented with a rad-hard technology, and therefore it is always free of errors. Whenever a failure is signaled, the checker takes control and corrects the problem. The concept in DIVA was extended in [16], considering that the checker itself could suffer errors. Therefore, when an error is detected, the recovery mechanism implies flushing the instruction pipe and resuming the execution at the instruction that produced the failure.

Another processor that follows the dynamic verification approach is Argus [17]. The idea behind this approach is that all von Neumann processors perform just four types of operations: choose the correct instruction to execute, perform the operation required by each instruction, forward results to the next instructions and interact with memory. In this way, the idea is that by checking these four groups of operations, a large number of errors can be detected by the architecture. To determine if the order of the instructions is correct, the system keeps a static control flow graph, which is dynamically recomputed in runtime. Any difference between the static and dynamic graph is considered an error. The same principle is used to determine if the correct results are forwarded to the next instructions. A static data-flow graph is kept and compared with the same structure dynamically recomputed. When both are different, the system determines that an error has occurred. Also, functional units are checked to verify that the operations determined by the executed instructions are correct. In some cases, this implies having a duplicate copy of the functional unit. In other cases, algorithmic properties of the operations are used to determine when an error has happened. Finally, in order to manage memory, the system checks if memory address calculations are correct and verify that the resulting addresses are properly aligned. Data integrity in the memory hierarchy is achieved with parity bits.

Not all the micro-architectural solutions are based on adding redundancy or special hardware to perform integrity checks. Another alternative is to organize information in such a way that the probability of suffering an error is decreased. For example, in [18], it is determined that the mentioned error probability is proportional to the time that instructions unnecessarily wait in unprotected structures. In this way, the strategy consists in minimizing the wait time in such structures, e.g., the instruction queue. When an instruction is in execution and it produces a data cache miss, the wait time of all subsequent instructions is considerably increased, as well as the probability that they can suffer an error. To mitigate this, those waiting instructions in the queue are invalidated and re-fetched later. In this way, they spend minimum waiting time and the probability of suffering an error decreases. Of course, this has a direct effect on delay and therefore the trade-off between performance and reliability should be explored. If the time saved by avoiding errors (and the associated recovery time) is larger than the delay overhead induced by invalidating and re-fetching instructions, then the strategy is reasonable.

1.3 Software Solutions

Reliability based on software techniques does not rely on adding hardware redundancy or implementing additional modules that can detect and correct errors. In this case, fault tolerance is achieved by modifying the software code running on the processor, so that possible errors can be detected at that level. In general, the cost overhead to implement these techniques is usually very low. However, this most of the times comes at a decrement of the system performance, since the processor has to execute longer programs, with more instructions than in the original code.

Many of the fault-tolerant software techniques are based on the so-called *temporal redundancy*. In this case, redundancy is not achieved by having several cores that execute the same program simultaneously, but by having a single core that executes the same program several times, in a sequential way. The program needs to be executed at least twice in order to detect errors. A discrepancy between both executions would imply that something has gone wrong. In order to correct the error, the program needs to be executed at least three times, majority voting the different executions if they produce incoherent results.

One of the best known fault tolerance software solutions is called SWIFT [19]. In this technique, software programs are modified by the compiler, which adds redundant instructions to achieve fault tolerance. Instructions are duplicated, although not identically, in order to avoid interferences between the two set of instructions. In this way, different registers are used by the compiler in the two sets, and separate memory spaces are kept for the load and store operations. Then, at certain points (fundamentally when a store operation saves a value in memory), both control flows are synchronized, comparing both results. If these results coincide, the system assumes that no error has happened. On the other hand, if both results are different, then, an error is signaled. Apart from the overhead introduced with the replication of instructions, this mechanism is not immune to errors. For example, if an error affects the operation code of an instruction, and for this reason that instruction changes into a store operation, the technique will not have control of the situation. This is due to the fact that the error has happened after the compiler finished its work, and therefore it would not be aware of the “new” store instruction.

There are several improvements that can be performed to extend the concepts in SWIFT, for example those in [20]. Three strategies are presented in this paper that complement the basic ideas in software reliability. The first strategy is called SWIFT-R, designed to allow recovery from errors. This principle behind this is straightforward, since it consists in adding two sets of redundant instructions instead of one. This can be seen as the software equivalent to TMR, rather than the DMR approach that the standard SWIFT would represent. This, many times, would introduce a too large overhead that would make the technique unfeasible. In order to keep the overhead smaller, another fault-tolerant technique, TRUMP, can be used. This technique only needs a set of redundant instructions, as SWIFT, but both the original and the redundant copies are not identical. The redundant copy is encoded based on AN-codes, that allow to determine which of the two copies is the one that has suffered the error in the case of discrepancy. AN-codes are a class of arithmetic codes, in which all the operands are multiplied by a constant value A. If this value is appropriately chosen, any single bit-flip on the operand would produce another value that is not AN-coded, thus determining that an error has happened. Therefore, the technique would work as follows. If both the original and redundant copies are different, an error in the system is signaled. If the redundant copy is still AN-coded, then the error has happened in the original copy. Otherwise, the error has happened in the redundant copy. This can be seen as a kind of diverse dual redundancy, since the intrinsic properties of each copy may indicate where the error has happened. The TRUMP technique has some limitations. First, multiplying all values by a constant makes that the actual range of values that can be stored in the registers smaller. Second, the AN-code transformation is not compatible with several arithmetic operations, thus posing a problem when data need to be processed in the functional units. Finally, there is another technique to increase software reliability, called MASK. In this technique, a semantic analysis is performed in the code, in order to identify those registers that contain narrow values, i.e., those whose most significant bits are always zero. These may occur for example in loop counters or in offset values associated to address calculations. When these values are identified, the compiler adds instructions before they are used in the code, to set all the upper bits to zero. In this way, if a bit-flip has affected one of those upper bits, it would be eliminated by forcing (masking) them back to zero.

Data integrity is not the only problem that affects software execution. Preserving the correct control flow is also important, since errors in the program counter usually produce undesirable effects. In [21], a method to check the correctness of the control flow is presented. The method is based on the generation of signatures, as a way of keeping track of the execution of the program. This is divided into basic blocks, and a signature is calculated for each of these blocks. This signature is stored in the code itself. In runtime, the actual execution is compared with the stored signatures. Any discrepancy between them is treated as an error in the control flow. This is a pure software method, since the comparison is achieved through some extra instructions that are added to the code. In this way, no extra hardware is needed to perform any of these operations.

Apart from the previously explained technique, there have been several subsequent alternatives that explore the possibilities of detecting control flow errors, as a way to improve reliability from the point of view of software. Some of these techniques can be found in [22–24]. Although these techniques are based on the same principles and rely on the manipulation of control flow graphs, all of them introduce a performance overhead due to the addition of the extra instructions needed to monitor execution. The addition of these instructions would be worth it if the incurred overhead is smaller than the error recovery time avoided by the technique. This idea is explored in [25], where the concept of *method efficiency* is introduced. This magnitude is defined as directly proportional to the number of errors detected by the technique, and inversely proportional to the performance overhead. In this way, different software-based techniques can be evaluated, finding out which ones offer better results for a given architecture and software program.

The techniques presented in this section have been designed to detect and correct soft errors, in which the effect produced by the event is temporary. Of course, these techniques can also handle permanent errors, but this is not their main purpose. For this kind of errors, there are other specific techniques that work more in the field of testability. Several of these techniques can be classified as SBST (*software-based self-testing*) [26, 27], an approach that has become accepted for microprocessor testing. The concept behind SBST techniques is to use the standard hardware resources in a microprocessor to run specific test programs, instead of having to add costly test-specific resources. With this approach, processors can be tested after production, in order to check for defects, but the same test programs can be run periodically, in production, to verify the state of the system. This approach allows nonintrusive operation (no extra hardware is needed), that can be run at the processor actual speed and can be applied in production, throughout the processor lifetime.

2 Solutions for Memories

Multicore processors rely on a hierarchy of memories to store data and instructions. Those include the register file, several levels of on-chip caches and the external memory. Each of those have different sizes and speeds being the ones closer to the cores faster but smaller. The use of a hierarchy also means that at a given time, there may be more than one copy of data or instructions at different levels. This is the case when a value is placed on a cache while the original value is also kept in the main memory.

The protection of memories has been widely studied in the literature and there are solutions to deal with manufacturing defects or permanent failures and with soft errors. In the first case, additional rows or columns are added to the memory array and they are used instead of failing elements when a defect is detected. This can be done at the production stage or in the field by executing self-test and self-repair mechanisms [28]. To protect against soft errors, ECCs are commonly used [29]. ECCs can also be used to protect against manufacturing defects but since those errors can be located and are permanent, replacing the failing memory cells with redundant ones is more effective. In this section, the focus is on the protection against soft errors that can occur at any time during the execution of a multicore processor using ECCs.

The ECCs used to protect memories are in most cases linear block codes [30] where the size of the block is selected to protect a memory word or a cache line. Convolutional codes are better suited for communications and there are only a few studies on their use to protect memories [31]. A block code takes a data block and computes a number of parity check bits that are stored with the data. This is done by an encoder circuit. Then when the data is read, the parity check bits are used to detect and correct errors. This is done by a decoder circuit. The use of an ECC introduces overheads as an encoder and decoder are needed and also additional memory bits need to be stored on the memory. The importance of those overheads depends on the memory being protected. For example, in a memory that needs to operate at very high speed the delay added by the encoder and the decoder may be the limiting factor while in other applications the memory size can be a constraint.

The complexity of the ECC depends also on the number of error patterns that have to be detected or corrected. In the simplest case where only single bit errors have to be detected, a parity check is enough while for the correction of multiple bit errors, advanced codes are needed. In memories, traditionally most errors have affected single bits and therefore Single Error Correction (SEC) codes are widely used [32]. However, as technology scales errors are more likely to affect several bits [33]. As those bits are physically close, the combination of a SEC code with interleaving can correct the errors. The problem is that interleaving makes the memory design more complex and requires an overhead. An alternative is to use ECCs that can correct adjacent errors [34, 35]. Another type of multiple errors is those caused by the accumulation of two different soft errors. Those are relevant in harsh environments where errors are frequent, like in space applications [36].

In the case of multicores, as there are different memories each with specific features the protection of each of them needs to be studied separately. At the same time, the interactions among the memories and their role in the multicore system need to be taken into account to achieve an optimal protection. In the rest of this section, the protection of each of the memories present in a multicore system is discussed outlining their specific requirements and the ECCs that can be used to protect them.

2.1 Register Files

In a multicore, registers are used on each core to store data and addresses that are used during the execution. The number of registers in a register file is small and they typically have the size of the processor architecture which in most modern multicores is 32 or 64 bits. The access to the register has to be fast and parallel as an instruction may operate on several registers at the same time. The speed requirement has traditionally limited the protection of register files to the use of a single parity check bit or a SEC code [37, 38]. Even the delay of single ECCs is in many cases too costly and several optimizations of SEC codes have been recently proposed to reduce the decoding delay [39, 40].The main issue with those codes is that in most cases they require additional parity check bits compared to a traditional SEC code. The replication of the register file is another option, but it requires triplication to implement error correction [41]. This large overhead can only be tolerated for applications where reliability is the critical requirement and area and power are not limited.

Other protection techniques for register files exploit the presence of narrow values in the register to detect and correct errors on narrow values using replication [42] or more sophisticated coding schemes [43]. These techniques can be useful to improve the reliability of register files but cannot provide a comprehensive solution as values that use the entire register are not protected. Alternative schemes have proposed the use of a shared error correction block that is used to protect only a subset of the registers focusing on those that are more critical based on their use [44, 45]. This selective protection scheme can be optimized if the compiler is aware of the register vulnerability and tries to maximize reliability [46]. However, as in the case of narrow values, only a fraction of the errors can be corrected.

As a summary to deal with single bit errors, for register files, at the hardware level the best solutions are to use a single parity check for error detection or a low delay SEC code for error correction.

In the case of multiple bit errors, the options to protect register files are limited as the use of more complex ECCs would impact speed. Some of the solutions presented make use of some form of replication where the replicas are placed in a different location so that they are not affected by the same particle hit [47]. Another option is to use interleaving that can also be problematic as it complicates the design of the register file. Finally, the parity bit or the SEC codes can also be replicated such that, for example, a per byte parity is used and the bits in each byte are physically apart. This can be seen as an interleaving at the ECC level that would also improve the speed as the encoders and decoders are faster for small block sizes. The main drawback of this approach is that the number of parity check bits required is also replicated, thus requiring a larger memory overhead.

As a summary, there are few options to deal with multiple bit errors on register files and most of them rely on some kind of replication or interleaving.

2.2 Caches

The next level in the memory hierarchy after the register file is the caches. These memories store copies of positions on the main memory that are currently in use [37]. In a multicore processor, there are typically several levels of cache that again have different size and speed. Level one (L1) caches are faster and smaller and are typically divided into two separate memories, one for data and another for instructions. L1 caches are included in every core and used exclusively by each core. Level two (L2) caches are larger but slower and are commonly shared by all cores and data and instructions are placed on a single memory. In some designs, there may be more than two levels of caches but in the following, the discussion assumes a multicore processor that has only two levels of cache.

There are several features that make caches different from a standard memory and that have implications for soft error protection. The first one is that by design cache stores copies of positions are stored in the main memory. Therefore, in many cases, there is a duplicated value in the main memory that can be used for error correction. The only exception is when a value stored in the cache has been modified and the copy stored in the memory has not been updated. Obviously, this is only possible when the cache stores data can be modified, but is not possible when the cache stores instructions. A direct consequence of this observation is that to protect L1 instruction caches, one option is to use an error detection code and once an error is detected it can be recovered by fetching the affected positions from the main memory. The same observation applies to data caches but only if write-through is used to manage writes to the cache. When data is written to the memory only when it is evicted from the cache, then the scheme can only be used for cache lines that have not been modified.

The second feature that makes a cache different from a standard memory is that several positions on the main memory map to the same position on the cache. Therefore, in addition to the value, a Tag that identifies the position in the main memory is needed on each entry. On a read access that Tag is compared to the upper bits of the address and only on a match the entry is used; otherwise, the cache has not the position requested and an access to main memory is needed. Since the Tags are stored in the cache, they can also suffer soft errors. A soft error on a Tag can create two types of effects: a false positive and a false negative. The first one occurs when an error changes the Tag and then an access is done to a position that matches the erroneous Tag. In this case, an incorrect value is used creating a situation that can lead to a system failure or to silent data corruption. On the other hand, a false negative occurs when after the error on the Tag, there is an access to the position that had the original Tag value. In this case, there will be no match and therefore the position will be fetched from the main memory. This can only create an error when the cache entry affected by the error on the Tag was modified.

A third feature that is specific to caches is that they commonly implement some type of associativity. That is, the cache has several ways and those are accessed all in parallel and the Tags are compared with the address to identify the way on which the value is stored. This implies that several ECC decoders are needed on the Tag memories to perform cache accesses.

A fourth feature is that caches also store control bits that mark, for example, the validity of a cache line, whether it has been modified and information on the use of the entry. From a soft error protection perspective, some of those bits have to be protected (valid and dirty bits) while other may not be so critical like, for example, the ones that track the use of the line.

Finally, a difference between caches and memories is that cache lines are typically wider as each line stores several positions on the main memories. For example, 512 bit lines are common in modern processors. This has also implications as the redundancy needed for an ECC depends on the block size and the same applies to the encoder and decoder complexity.

There are several options to protect caches, both for the data part and for the tag and control information. In all cases, speed is an important requirement as is the memory size since caches need to be fast and its size is an important fraction of the processor area.

Focusing on the cache entries themselves, in the case of instruction caches as discussed before, on an error, the data can be recovered from the main memory. Therefore, an error detection code is enough and a single parity check bit can be used to detect single bit errors. In the case of multiple bit errors, a SEC code can detect double bit errors and a Single Error Correction Double Error Detection (SEC-DED) can detect triple bit errors. This means that for most practical scenarios a parity check or a SEC/SEC-DED code will be sufficient. The same analysis applies to the data caches when a write-through policy is used [48]. However, in many cases using write-through requires significant overheads in terms of power and memory bandwidth.

When a write-back policy is used in a data cache, errors have to be corrected on dirty lines to avoid data corruption. This can be done for single bit errors using a SEC-DED code or more advanced codes that can correct multiple bit errors [49]. Other schemes based on replication of the data have also been proposed [50] but they either require a large area overhead or provide only partial protection. If correction is done on every access, then low delay codes such as Orthogonal Latin Squares codes can be used [51]. Those codes require a larger number of parity check bits than other multibit ECCs but the error correction procedure is much simpler. This has motivated their use in caches [52]. An interesting option is to split the error protection on two parts, error detection and error correction. When that is done, the error correction part of the ECC can be placed outside the cache as proposed in [53] to minimize the cost. In any case, the scheme significantly reduces the overheads required for the read accesses that are error-free as a full decoding is not needed. The split between the error detection and error correction parts of the ECC can also be optimized and more complex codes such as Bose–Chaudhuri–Hocquenghem (BCH) codes can be used [54].

For the protection of the Tags, similar considerations as those discussed for the values apply. In the case of instruction caches, detecting the errors to avoid false positives that can lead to the execution of erroneous instructions is enough. The same applies data caches that use a write-through policy. For data caches that use write-back, errors need to be corrected when the cache line is dirty. In this case, a SEC or SEC-DED code can be used to correct single bit errors. Several works have noted that in many cases, nearby entries on the cache have the same Tag. This inherent redundancy can be used to implement error correction by using another copy of the Tag when an error is detected [55]. Replication can also be introduced in the implementation to improve reliability [56].

As noted before, a difference of Tags is that they are accessed in parallel to check for a match in all the ways of the cache. This means that in a direct implementation, a decoder is needed for each way. To reduce this overhead, an alternate is to encode the Tag value of the access using the ECC and then do a comparison with the encoded Tags that are stored in the cache. This scheme is known as FastTag [57]. When the Tags are protected with a parity bit, an exact match comparison can be used to avoid false positives as illustrated in Fig. 4. However, when a SEC code is

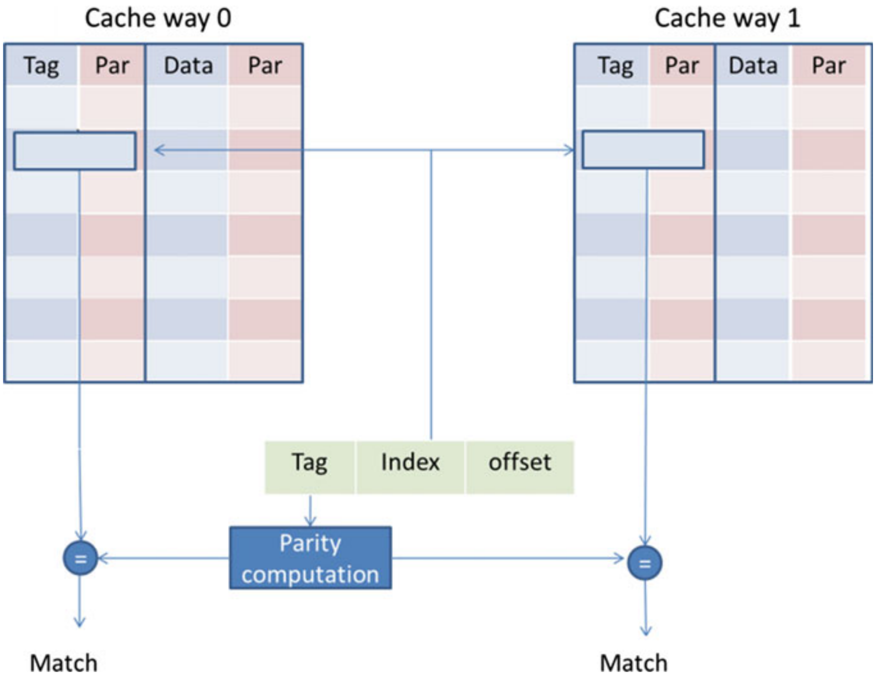


Fig. 4 FastTag implementation using a parity bit (from Ref. [57])

used, distance-one comparators are needed as illustrated in Fig. 5. In both cases, the scheme can be useful to reduce the latency and area required to implement error detection or correction on the Tags.

Finally, for the protection of the control flags such as the valid bit or dirty bit there are also several options. They can be protected with the same ECC that protects the Tag that as discussed before depends on the type of cache. In some cases, it may be of interest to provide additional protection for these flag. For example, if a SEC-DED code is used, it can be of interest to be able to implement double error correction for the flags only so that when a double error occurs, we can determine if the cache line was valid or dirty. When the line is invalid, the error can be ignored and when it is clean the error can be recovered by fetching the copy stored in main memory. Therefore, in both cases, the error is effectively corrected. The protection of the flags bits can be done by extending existing SEC-DED codes as proposed in [58].

2.3 Main Memory

The last level of the memory hierarchy in a multicore processor is the main memory that is external to the processor and typically composed of several memory devices.

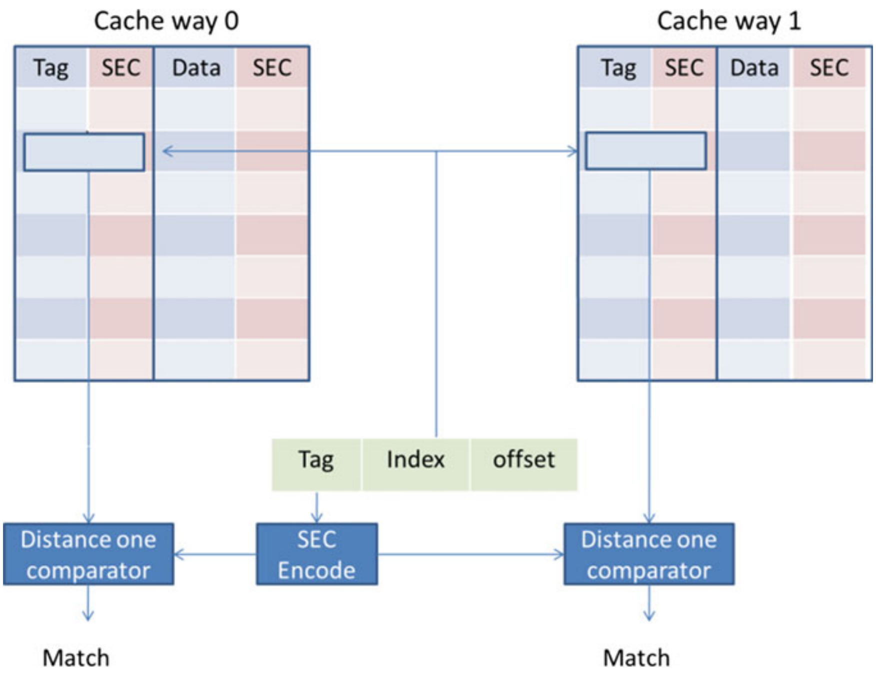


Fig. 5 FastTag implementation using a SEC code (from Ref. [30])

The main memory is large but slow as off-chip accesses are orders of magnitude slower than on-chip accesses. The dominant technology for external memory is DRAM that enables large capacity with a reasonable cost. The memories used support in most cases a burst mode where the access to a number of consecutive positions is much faster than accesses to unrelated positions. This mode fits well with the use of large cache lines that comprise several memory words. To maximize the bandwidth of the memory interface, advanced methods such as the different generations of Double Data Rate (DDR) are commonly used. This means that the memory devices need to incorporate complex control logic that is also susceptible to soft errors.

The errors on the DRAMs have been studied in the field during their operation in servers [59, 60] as they are an important issue for large data center and high performance computing facilities. They have also been evaluated in several accelerated radiation tests [61]. In both cases, the results show that in addition to soft errors affecting the memory cells, there are Single Event Functional Interrupts (SEFIs). Those occur, for example, when a soft error affects the memory controller and it stops working properly. A SEFI can affect an entire memory device or only part of it, but in all cases, the data coming from that device can be erroneous. Therefore, the protection for DRAMs in a multicore processing system should be able to cope with the failure of one of the memory devices. To this end, several solutions have been proposed over the years based on the use of several interleaved codes, Reed Solomon Codes or modified SEC-DED codes [62–64]. All those solutions support the failure of one of the devices and are commonly referred to as “Chipkill” solutions [62].

Recent works, have proposed several alternatives to improve the protection of the external memory. Some of them focus on the construction of ECC schemes that provide flexibility to place parts of the parity check bits on different locations [65, 66]. For example, the bits needed to perform error detection can be placed with the data while the rest of the parity check bits can be placed elsewhere as they only need to be accessed in case of error. As errors are rare, this enables the use of stronger ECCs with little overheads for read operations. The main limitation of this type of schemes is that all the bits need to be accessed for write operations. Another related idea is to use the flexibility provided by having the parity check bits in a different location to implement a not uniform error protection depending, for example, on the criticality of the data. All these schemes combine the use of ECCs with aspects of the memory organization on a computer system. In many cases, this has implications on performance, for example, by penalizing write operations in the case of using different locations for the parity check bits used for detection and correction. The focus of this section is on the ECCs that can be used for each memory in a computing system, rather than on the interactions of the memory with the rest of the computing system and its organization. Readers interested in exploring this topic further can find detailed information in the references provided in [65, 66].

Other area of research is how to exploit the mapping of the device bits to optimize and refine the error correction scheme [67]. In all cases, the main restriction is the memory architecture that fixes the width of the memory interface and the number of memory bits that are available in addition to the data bits. Those are commonly eight bits for every sixty four data bits. The use of different locations, enable the use of entire words or lines for error correction giving more flexibility at the cost of performance degradation.

Finally, the emergence of 3D DRAM memories will have implications for the protection techniques. First, the errors will most likely be different, and second the memory organization and structure may also change [68]. In fact, the use of 3D memories may have implications for other parts of the processing system such as the caches [69]. This will open new research areas in which reliability and fault tolerance will play an important role.

3 Solutions for Interconnections

Several aspects make important to develop ad hoc solutions for increasing dependability of interconnections in multicore architectures.

The first aspect is the part of the multicore chip dedicated to this specific resource. The design of a multicore processing system poses several challenges in the definition of the on-chip interconnection architecture needed for transfer of data between the cores and the shared memories (e.g., the shared L2 cache) and for direct inter-core communication. With the current technology nodes, the design of the interconnection fabric cannot be carried out independently, but a joint co-design of cores, memories and interconnection mechanisms is needed. This is due to the high design constraints (power, area, latency, and bandwidth) required for on-chip interconnects and also for the effect that the interconnection design has on the rest of the system [70]. Therefore, the interconnections became a critical design element and their requirements in terms of are and power cannot be neglected. As stated in [70], even in a conservative scenario, the power cost of the interconnections can be estimated as equivalent to one core of an eight cores chip, and the area cost as equivalent to three cores.

The second aspect to take into account is the energy consumption required by the global wires and the request of low-voltage signaling for on-chip communication [71]. While low swing signaling can provide one order of magnitude power savings, also reduce the noise margin and greatly increase the probability of error occurrence in the data transmission.

Finally, another aspect to take into consideration is the type of faults that can affect the interconnections. While permanent faults will be mostly due to electromigration mechanism affecting the wires composing the interconnections, the transient faults are mostly due to electrical noise and to cross-talk effects.

The methods to increase the dependability of the multicore interconnections depends on the specific interconnection topology, and are usually based on modifying existing methods developed for other scenarios (e.g., from the radio transmission) or ad hoc methods to face specific problems (e.g., the cross-talk induced errors). In the rest of this section, we will describe the most common interconnection topologies and the most used dependability methodology.

3.1 Interconnection Topologies

Different topologies have been proposed to connect together multicore architectures [71]. Some of these topologies are based on shared bus communication infrastructures, often connecting together private memories (e.g., L1 private caches are connected to a shared L2 cache). When bus-based topologies are used, error codes are the most viable solution. In particular, a naive solution is to use the same ECC used in memory, also for data movement. Despite the simplicity of this approach, the achievable performance can be not optimal, due to several factors. The different width of data bus and memory array can require different code rate, while the different type of errors that can occur during storage and transmission can be faced efficiently by different type of codes. As an example, interleaving is widely used in memories to prevent multiple adjacent bit upset, but when a codeword is fetch from the memory, it is no more protected against this type of errors.

With the increased number of cores, the Network-on-chip (NoC) approach became a viable solution for interconnection topologies to provide flexibility, scalability and high performance [72]. Figure 6 shows the typical NoC

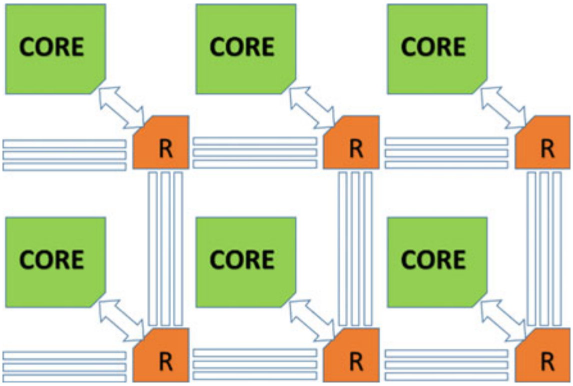


Fig. 6 Typical NoC interconnection structure

interconnection, with the cores connected using the routing blocks that manage the data exchange among the cores.

The uses of a NoC based interconnection allow using more error tolerant techniques, since error can be handled at link level or at network level [73, 74]. A very similar approach is used also when NoC topology are implied.

The data integrity of on-chip interconnection can be enhanced working at layout level, for example using suitable space rules, shielding, etc., or at electrical level, using signal repeaters, low noise buffers, etc. Unfortunately, these solutions are strictly technology dependent and are very sensitive to the technology shrink. Therefore, a more general approach that can be widely applied to the problem of interconnection data integrity is needed; the approach that more is based on information redundancy. The additional information is used to detect data errors, and, depending on the specific method used, to recover the corrected data. Recovery can be achieved using retransmission, as in Automatic Repeat Request (ARQ) or using ECCs for Forward Error Control (FEC). Other methods use hybrid ARQ/FEC schemes. The above-mentioned schemes are mainly designed to handle transient faults, even if the use of ECCs can also tolerate permanent faults if the error activated by the permanent fault does not exceed the error correction capability of the code. Techniques to tolerate permanent faults require the use of redundant interconnection resources. These techniques have been widely studied in the case of NoC interconnection. In particular, since NoC architectures inherently provide multiple paths from the source to the destination cores, it is possible to develop fault-tolerant routing algorithm to handle permanent faults in the interconnection structure of NoC based multicores.

3.2 *Automatic Repeat Request (ARQ)*

ARQ is based on the retransmission of erroneous data. This method provides very good performance for low error rate, as stated in [73]. As anticipated, the information redundancy is used to detect errors in the received data. The receiver send to the sender a request to retransmit data is an error is detected. It is worth to notice that, even if low-voltage signaling is used to send data to the received, the return path is usually exploit full voltage swing, to avoid complexity/issues related to the data integrity of the ACK signals [73].

Three main implementation of ARQ methods for on-chip interconnection have been proposed [75]: *stop-and-wait*, *go-back-N*, and *selective-repeat*.

The simplest but less performing method is *stop-and-wait*, in which after sending a data, the sender waits for the acknowledgement (ACK) from the receiver. This method has a very low throughput and is not well suited for high-speed communications. In the *go-back-N* method the sender store up to N data samples and send it to the receiver. The window size N directly related to the signal rounds trip to permit the continuous streaming of data, superseding the limitation of the *stop-and-wait* ARQ. When the received detect an error, it request to the sender to transmit again the last N data. Despite the increased complexity with respect to the previous method, the throughput gain make this method appealing for on-chip communication. Finally, *selective-repeat* add to the previous method the property of selecting which only the erroneous received data among the N data in the transmission window is resend.

3.3 Forward Error Control (FEC)

FEC uses information redundancy to correct errors in the received data, avoiding the retransmission cost. The drawback of this method is the additional resources needed to transfer the redundant information from the sender to the receiver. Moreover, the data transmission blocks require ECCs encoders and decoders. These elements are a further element of overhead.

On the other hand, FEC schemes provide a fixed throughput, a simpler communication protocol and better performances with respect to ARQ when the error rate increases. Among the different types of ECC, few of them have the right characteristics for realizing on-chip FEC communication. In this section, we will focus on single error-correcting (SEC) codes based on Hamming codes, Orthogonal Latin Squares codes and Crosstalk Avoidance Codes (CAC). This last code is particularly suitable to avoid the problems related to the interconnection capacitive coupling.

3.3.1 Hamming Codes

Hamming codes among the most used error codes. The code takes as input a k bits dataword and provide $(n-k) = [\log_2(k)]$ check bits to form a n bits codeword. The encoded and decoder are based on a combinatorial network of XOR gates and can be easily implemented with a limited amount of resources. The decoder is able to correct any single bit error. This code is effective against single bit errors and the limited number of additional wires required to transport the additional check bits make this code very appealing for on-chip interconnection FEC schemes.

3.3.2 Orthogonal Latin Squares Codes

In [76] Orthogonal Latin Square Code (OLSC) has been proposed to provide protection against multiple bit errors for on-chip interconnections. Since OLSC are one-step-decodable majority code, it can be decoded with a very high speed. Moreover, the modularity of the check matrix allows easily adapting the error correction capability of the code depending on the expected error rate and on the available hardware resources. The number of check bits for an OLS code with a dataword of k bits is $(n-k)=2t\sqrt{k}$, where n is the codeword length and t is the number of correctable errors. The information redundancy required by the OLS code with $t = 1$ is higher than the one of the Hamming code, and similarly, the OLS code with $t > 1$ has higher redundancy than other multibit ECCs, such as BCH codes. However, the complexity and the cost of encoding/decoding BCH codes prevent their use in on-chip interconnection FEC. Moreover, the use of OLS code with $t > 1$ can be attractive with respect to the SEC Hamming code, since the interconnection can operate at lower voltage level. Thus, even if the number of wires increase with respect to the Hamming code, the energy spent in transferring the singe bit is much less, providing a significant overall energy saving.

3.3.3 Crosstalk Avoidance Codes

The maximum frequency of on-chip data transmission is related to the worst case switching capacitance of the metal wires. In case of adjacent wires, this worst case occurs when a wire switches in one direction and the two adjacent wires switch in the opposite direction. This capacitance can be estimated as $C_{wc} = (1 + 4\lambda)\ C_L$, where C_L is the load capacitance and λ is a parameter that take into account the influence of the adjacent wires.

The idea behind the CAC is to reduce this capacitance using information redundancy to avoid the occurrence of the worst case.

A first attempt to exploit this idea is given in [77], where Hamming codes and Dual Rail (DR) codes are compared with respect to their ability to reduce the crosstalk effect. DR codes, also known as Duplicate and parity (DAP) codes, are SEC codes composed by two copies of the data bits and by a parity check bit formed by the xor of all the data bits. The corrected word can be retrieved selecting which of the two copies of the data word present the right parity. The paper also discusses the use of intelligent spacing, in which the distance between adjacent wires carrying identical values differs from the distance of adjacent wires carrying different values. The delay reduction achievable by intelligent spaced DR codes is more than 30% with respect to the standard Hamming code.

The authors of [78] argues that since any linear ECC for minimization of crosstalk effects requires two times the number of data bits, it is possible to design joint crosstalk avoidance and multiple ECCs (CAC/MEC) duplicating a SEC codeword. Going into details, it is possible to duplicate a SEC-DED Hamming or Hsiao codes to provide triple error correction. The rationale of the CAC/MEC decoder is similar to the DR decoder. It selects between the two copies the SEC-DED codeword the one that have 0 errors or only 1 error and, if needed, perform the SEC correction on this codeword.

Finally, we mention the method proposed in [79], where first a nonlinear ECC code is used to minimize the crosstalk effect, and after this code is embedded in a linear ECC code such as the standard Hamming code.

3.4 Hybrid ARQ/FEC

An hybrid method to protect data transmission can be designed combining the ARQ and the FEC methods. The ides of the hybrid ARQ/FEC (HARQ) data protection scheme is to transmit the data using an ECC able to correct up to n errors and to detect up to $n + m$ errors. If the number of errors that affect the transmitted codeword is less or equal to n, the method act as the FEC scheme. If up to $n + m$ errors occur to the codeword, the system will ask to retransmit the codeword. This method can provide higher reliability than error control with FEC alone and higher throughput than error control with ARQ alone.

Among the different types of hybrid ARQ/FEC we mention the subdivision in type-I HARQ and type-II HARQ. Type-I HARQ requires the transmission of the whole codeword, and the codeword is retransmitted if the number of detected errors exceeded the maximum number of correctable errors [75]. Instead, in the type-II HARQ method, if the number of detected errors exceeded n, the system send additional check bits in order to increase the code correction capability. The type-II HARQ therefore require sending less data with respect to type-I in the retransmission phase. type-II HARQ methods are based on Hamming codes [80] or on extended Hamming product codes [81].

3.5 Fault-Tolerant NoC Interconnection

NoC architectures transmit data among the different processing elements using specific hardware blocks called routers. Each routers has a local port connected with his processing element and several ports connected with other routers. The transmission of data from a processing element to another one occurs traversing several routers. If one of the routers is faulty, or if one of the link connecting two routers is faulty, the data must be transmitted avoiding the faulty resource. This mechanism provides a graceful degradation behavior of the network, which can works even in presence of permanent faults, both with degraded performance (e.g., in terms of throughput or latency). When the faulty elements break the network in two or more disjoint networks, the system goes in an unrecoverable faulty state. A fault-tolerant NoC interconnection structure should be able to define a transmission path between any two processing elements, if this path exists. There are two mechanisms to achieve fault tolerance in NoC interconnections [82]. The first is based on the use of redundant packets, the second on the use of redundant routes. Redundant packets mechanisms reply the incoming packets and transmit on multiple links, increasing the probability that a copy of the packet will arrive to the final destination. Redundant routing algorithms exploit information on the network status (link/router failures) to decide the output port of the incoming packets. We remark that, using a suitable error detection code, the first method is also useful to tolerate transient faults. In fact, a router can discard a corrupted copy of a transmitted packet, relying on the fact that other copies of the same packet have been transmitted. Instead, the redundant route based methods must implement a specific method to face the occurrence of transient faults. For example, in [83] a hybrid FEC/ARQ scheme is used for transient fault, while a hierarchical routing table based scheme is proposed to handle permanent faults.

3.5.1 Fault Tolerance Using Redundant Packets

Fault-tolerant algorithms based on redundant packets reply each packet transmitted by a sender to increase the likelihood of reaching the destination. The simplest mechanism is to broadcast each packet received by a router to all the other output ports of the router [84]. This approach guarantees that the packet is transmitted among all the possible paths from the source to the destination. Unfortunately, the transmission overhead of this method can become very huge when the network size grows. In fact, if the broadcasting is applied to all the incoming packets, the number of replicas grows exponentially, and this method can suffer for network congestion effects. To mitigate this effect, several methods have been proposed. Some of them copy the incoming packets with probability p, thus limiting the number of replica traveling in the network [85]. This probability can be weighed by the distance from the current router to the destination, to drive the wave of packets toward the destination [86]. All the method proposed in literature focuses on limiting the number of copies to transmit in the network. The power consumption and the throughput overhead of these methods are in fact directly proportional to the number of number of packet copies. However, this overhead is always present in the network, also when he operate in a fault free condition. Instead, the methods based on redundant routes only degrade the network performances when a fault is actually present.

3.5.2 Fault Detection

The use of redundant routes to tolerate faults requires the identification of the faulty NoC resources (the routing blocks or the link interconnections) and rerouting the packets to avoid the use of these resources. The first step is, therefore, the detection of faults occurring in the NoC infrastructure. Two methods have been proposed to detect faults occurring in these elements: (1) exploiting the information gathered using the method to tolerate transient faults and (2) applying built-in self-test procedures to identify the faulty resources.

The first method can use the methods previously described (ARQ, FEC, HARQ) to detect the occurrence of permanent faults. When a packet is not received, or is corrupted, the error can be due to a transient or to a permanent fault. When the same resource experiences a too high number of errors, it can be detected as faulty. Example of this approach can be found in [83], where a detailed fault diagnosis process is described. The main drawback of this technique is the fault diagnosis granularity. For example, if an end-to-end ARQ communication method is used, the fault detection procedure can only identify a faulty path, without getting direct information on which element of the fault is broken. Suitable algorithms to increase the fault location capabilities must be developed to identify the faulty resource.

The other method for fault detection in the NoC infrastructure is based on the use of an on-line testing procedure. This test procedure are based on the classical Built-in Self-Test (BIST) method, that requires a test pattern generator (TPG) and output response analyzer (ORA) to stimulate the element under test and to analyze the response to detect a faulty behavior. In [87] this approach has been used in conjunction with redundant wires to provide non-interrupted on-line testing. The testing routine sequentially selects the wires to test, rerouting the data that should be delivered by these lines to spare wires. Due to the large number of elements to test, it is important to reduce the time to test (applying the test vectors in parallel, or reducing the number of test patterns) and the resource needed to perform the test (e.g., sharing the TPG or the ORA blocks). The authors of [88] propose to exploiting the structure of the NoC to achieve test parallelism. The test patterns are organized as test packets and delivered to the element under test exploiting the NOC infrastructure. In this way, it is possible share the TPG/ORA blocks and the Test Access Mechanism (TAM) ports for delivering test data to the components under test.

The two described methods can be used in conjunction, as proposed, for example, in [83, 87]. The detection of repeated transient error triggers an on-line testing procedure that precisely identify the faulty resource.

3.5.3 Fault-Tolerant Routing Algorithms

There is a large variety of fault-tolerant routing algorithms, which differs for several aspects. Following the differentiation proposed in [89], where a comprehensive survey of fault tolerance for NoC is presented, we divide the algorithms depending on the locality degree of the fault information. The routing algorithms that exploit global information can easily provide the shortest path to the packets even in presence of faults, and can avoid deadlocks. It is important to notice that fault-tolerant routing algorithms should guarantee deadlock—free routing also in presence of faults. Routing algorithms based on global information use some sort of routing table inside each router that can be reprogrammed when a fault is detected. Examples of algorithms based on global information are [90], where the minimization of the routing table size is faced, and [91], where a region-based routing algorithm is proposed. The main drawback of the algorithms that use global information is the hardware and software complexity.

In contrast, algorithms based on local fault information of the switch itself and its immediate neighborhood are easier to implement, but must explicitly manage the problems related to deadlock. These routing algorithms modify the path transporting the packets using the neighbors of a faulty router. Techniques to avoid deadlock can be based on routing restrictions (in [92] packets are routed around the fault location along the contour in which two turns are prohibited), or using additional packet header information (see, e.g., [93]).

Other methods for fault-tolerant routing algorithms exploit the use of virtual channels [94–96]. Virtual channels can be created on the top of the network infrastructure by sharing the same physical channels using packet based multiplexing. Originally proposed for off-chip networks [97], virtual channels can divide circular traffic into different channels to avoid deadlocks. It must be noticed virtual channels can be created only adding suitable queues, multiplexing and control logic to the routers. This requires a considerable amount of logic resource and consequently increases the power budget.

Finally, inherently fault-tolerant routers can be designed, that use internal redundant resources (FIFOs, crossbar, etc.) to tolerate faults. Examples of these hardened routers are presented in [98, 99]. These redundant resources can be used also in conjunction with suitable routing algorithms, as proposed in [100].

4 Summary

This chapter has discussed existing solutions to detect and correct errors in the different parts of a multicore system.

About processors, several techniques have been covered from the architectural, micro-architectural and software points of view. Architectural solutions mostly rely on redundancy, using several cores to run the same process and compare the outputs through a voter. Recovery mechanisms are complemented with the use of standard techniques, as checkpointing. On the other hand, micro-architectural approaches tend to apply more heterogeneous solutions to the different parts of the processor. For example, critical areas may be protected using massive triplication, while other less critical parts can be periodically checked for errors (e.g., checking for system invariants). Finally, software approaches are, many times, based on redundancy too.

But since this redundancy is applied to the software program and not the hardware itself, the area overhead is reduced at the expense of performance degradation.

In the case of memories, most of the solutions are based on the use of ECCs or replication. However, each memory has different features and requirements. For example, speed is critical in a register file and therefore the decoding complexity that can be tolerated is limited, while for a DRAM speed is much lower and more complex decoders can be used. This means specific solutions are used at each level in the memory hierarchy. In addition, the interaction between levels is also important. For example in the case of caches, where the use of the copy stored in main memory can in many cases be useful to recover from errors.

Different techniques to handle the interconnection part have also been provided, with a special emphasis on NoC. In this way, some detection and correction solutions for this kind of technology have been presented, including ARQ and FEC.

Finally, this chapter only aims at presenting an overview of the existing solutions. The reader can find more information in the references to dig deeper into any particular aspect of the different alternatives to protect a multicore system.

References

1. H. Cho, S. Mirkhani, C.-Y. Cher, J. Abraham, S. Mitra, Quantitative evaluation of soft error injection techniques for robust system design, in *Proceeding of DAC'13*, Austin, TX, USA
2. M. Ottavi, S. Pontarelli, D. Gizopoulos, C. Bolchini, M.K. Michael, L. Anghel, M. Tahoori, A. Paschalis, P. Reviriego et al., Dependable multicore architectures at nanoscale: the view from Europe. IEEE Des. Test Comput. **32**(2), 17–28 (2015)
3. S.K. Reinhardt, S.S. Mukherjee, Transient fault detection via simultaneous multithreading, in *Proceedings of The 27th International Symposium on Computer Architecture*, June 2000
4. S.S.Mukherjee, M. Kontz, S.K. Reinhardt, Detailed design and evaluation of redundant multithreading alternatives, in *ISCA*, 2002
5. S. Campagna, M. Hussain, M. Violante, Hypervisor-based virtual hardware for fault tolerance in COTS processors targeting space applications, in *Proceedings of International Symposium on Defect Fault Tolerance VLSI System*, 2010, pp. 44–51
6. D. Gizopoulos et al., Architectures for online error detection and recovery in multicore processors, in *Proceedings of Design, Automation Test in Europe* (*DATE*), 2011, pp. 533–538
7. J.T. Daly, A higher order estimate of the optimum checkpoint interval for restart dumps. Future Gener. Comput. Syst. **22**(3), 303–312 (2006)
8. M. Bougeret, H. Casanova, M. Rabie, Y. Robert, F. Vivien, Checkpointing strategies for parallel jobs, in *Supercomputing, SC '11* (ACM, New York, NY, USA, 2011), pp. 1–11
9. X. Ni, E. Meneses, N. Jain, L.V. Kalé, ACR: automatic checkpoint/restart for soft and hard error protection, in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, Denver, Colorado, 17–21 November 2013
10. B. Mills, R. Melhem, Shadow computing: an energy-aware fault tolerant computing model, in *2014 International Conference on Computing, Networking and Communications* (*ICNC*), Honolulu, HI, 3–6 February 2014
11. H. Zhou, A case for fault-tolerance and performance enhancement using chip multiprocessors. IEEE Comput. Archit. Lett. **5**(1), 22–25 (2006)

12. H. Zhou, Dual-core execution: building a highly scalable single-thread instruction window, in *PACT'05*, 2005
13. J. Gaisler, A portable and fault-tolerant microprocessor based on the SPARC v8 architecture, in *Proceedings of International Conference on Dependable Systems and Networks*, 2002, pp. 409–415
14. L.T. Clark, D.W. Patterson, C. Ramamurthy, K.E. Holbert, An Embedded Microprocessor Radiation Hardened by Microarchitecture and Circuits. IEEE Trans. Comput. **65**(2), 382–395 (2016)
15. T.M. Austin, DIVA: a reliable substrate for deep submicron microarchitecture design, MICRO 1999
16. A. Bouajila, T. Sommer, J. Zeppenfeld, W. Stechele, A. Herkersdorf, A Fault-Tolerant Processor Architecture, in *22nd International Conference on Architecture of Computing Systems* (*ARCS*) (Delft, The Netherlands, 11 March 2009), pp. 1–5
17. A. Meixner, M.E. Bauer, D.J. Sorin, Argus: low-cost, comprehensive error detection in simple cores, MICRO (2007)
18. C. Weaver, J. Emer, S. Mukherjee, S.K. Reinhardt, Techniques to reduce the soft error rate of a high-performance microprocessor, in *Annual International Symposium on Computer Architecture*, 2004
19. G. Reis, J. Chang, N. Vachharajani, R. Rangan and D. August, SWIFT: Software implemented fault tolerance, in Proceedings of International Symposium on Code Generation Optimization, 2005, pp. 243–254
20. G.A. Reis, J. Chang, D.I. August, Automatic instruction-level software only recovery method. IEEE Micro **27**(1) (2007)
21. N. Oh, P.P. Shirvani, E.J. McCluskey, Control flow checking by software signatures. IEEE Trans. Reliab. **51**, 111–122 (2002)
22. Z. Alkhalifa, V.S.S. Nair, N. Krishnamurthy, J.A. Abraham, Design and evaluation of system-level checks for on-line control flow error detection. IEEE Trans. Parallel Distrib. Syst **10**(6), 627–641 (1999)
23. O. Goloubeva, M. Rebaudengo, M.S. Reorda, M. Violante, Soft-error detection using control flow assertions, in *Proceedings of 18th IEEE International Symposium Defect and Fault Tolerance in VLSI Systems*, 2003, pp. 581–588
24. R. Venkatasubramanian, J.P. Hayes, B.T. Murray, Low-cost on-line fault detection using control flow assertions, in *IOLTS'03: Proceedings of 12th IEEE International On-Line Testing Symposium*, 2003, pp. 137–143
25. R. Vemu, J. Abraham, Ceda: control-flow error detection using assertions. IEEE Trans. on Comput **90**(9), 1233–1245 (2011)
26. M. Psarakis, D. Gizopoulos, E. Sanchez, M. Sonza Reorda, Microprocessor software-based self-testing. IEEE Des Test of Comput **27**(3), 4–19
27. N. Foutris, M. Psarakis, D. Gizopoulos, A. Apostolakis, X. Vera, A. Gonzalez, MT-SBST: self-test optimization in multithreaded multicore architectures, in *Proceeding of IEEE Internationl Test Conference*, 2010, pp. 1–10
28. J.-F. Li, J.-C. Yeh, R.-F. Huang, C.-W. Wu, A built-in self-repair design for RAMs with 2-D redundancies. IEEE Trans. Very Large Scale Integr. Syst. **13**(6), 742–745 (2005)
29. C.L. Chen, M.Y. Hsiao, Error-correcting codes for semiconductor memory applications: a state-of-the-art review. IBM J. Res. Dev. **28**(2), 124–134 (1984)
30. S. Lin, D.J. Costello, *error control coding*, 2nd edn. (Englewood Cliffs, New Jersey, Prentice-Hall, 2004)
31. J.J. Metzner, Convolutionally encoded memory protection. IEEE Trans. Comput. **31**(6), 547–551 (1983)
32. M.Y. Hsiao, A class of optimal minimum odd-weight column SEC-DED codes. IBM J. Res. Dev. **14**(4), 395–401 (1970)
33. E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, T. Toba, Impact of scaling on neutron-induced soft error rate in SRAMs from a 250 nm to a 22 nm design rule. IEEE Trans. Electron Devices **57**(7), 1527–1538 (2010)

34. A. Dutta, N.A. Touba, Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code, in *25th IEEE VLSI Test Symposium*, 2007, pp. 349–354
35. L.J. Saiz-Adalid, P. Reviriego, P. Gil, S. Pontarelli, J.A. Maestro, MCU tolerance in SRAMs through low redundancy triple adjacent error correction. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **23**(10), 2332–2336 (2015)
36. M.A. Bajura et al., Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs. IEEE Trans. Nucl. Sci. **54**(4), 935–945 (2007)
37. S. Mukherjee, Architecture design for soft errors (Morgan Kaufmann, 2008)
38. E. Fetzer, D. Dahle, C. Little, K. Safford, The parity protected, multithreaded register files on the 90-nm Itanium microprocessor. IEEE J. Solid-State Circuits **41**(1), 246–255 (2006)
39. P. Reviriego, S. Pontarelli, J.A. Maestro, M. Ottavi, Low-cost single error correction multiple adjacent error correction codes. IET Electron. Lett. **48**(23), 1470–1472 (2012)
40. P. Reviriego, S. Pontarelli, J.A. Maestro, M. Ottavi, A method to construct low delay Single Error Correction (SEC) codes for protecting data bits only. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **32**(3), 479–483 (2013)
41. R. Naseer, R. Bhatt, J. Draper, Analysis of soft error mitigation techniques for register files in IBM Cu-08 90 nm technology, in *Proceeding of IEEE International Midwest Symposium Circuits and Systems*, 2006, pp. 515–519
42. O. Ergin, O. Unsal, X. Vera, A. González, Exploiting narrow values for soft error tolerance. IEEE Comput. Archit. Lett. **5** (2006)
43. I.B. Karsli, P. Reviriego, M.F. Balli, O. Ergin, J.A. Maestro, Enhanced duplication: a technique to correct soft errors in narrow values, IEEE Comput. Archit. Lett. **12**(1), 13–16 (2013)
44. P. Montesinos et al., Using register lifetime predictions to protect register files against soft errors, in *Proceeding of Dependable Systems and Networks*, 2007, pp. 286–296
45. J. Lee, A. Shrivastava, Static analysis to mitigate soft errors in register files, in *Proceeding of Design, Automation and Test in Europe* (*DATE*), April 2009, pp. 1367–1372
46. J. Lee, A. Shrivastava, A compiler-microarchitecture hybrid approach to soft error reduction for register files. IEEE Trans. Comput. Aided Des. Integr. Circuits and Syst. **29**(7), 1018–1027 (2010)
47. M. Fazeli, A. Namazi, S.G. Miremadi, An energy efficient circuit level technique to protect register file from MBUs and SETs in embedded processors, in *Proceeding of Dependable Systems and Networks*, 2009, pp. 195–204
48. L.T. Clark, D.W. Patterson, C. Ramamurthy, K.E. Holbert, An embedded microprocessor radiation hardened by microarchitecture and circuits. IEEE Trans. Comput. **65**(2), 382–395 (2016)
49. J. Kim, N. Hardavellas, K. Mai, B. Falsafi, J.C. Hoe, Multi-bit error tolerant caches using two-dimensional error coding, in *Proceeding of the 40th IEEE/ACM International Symposium on Microarchitecture* (*MICRO*), December 2007
50. W. Zhang, S. Gurumurthi, M. Kandemir, A. Sivasubramaniam, ICR: In-cache replication for enhancing data cache reliability, in *Proceeding of the International Conference on Dependable Systems and Networks* (*DSN*), June 2003
51. M.Y. Hsiao, D.C. Bossen, R.T. Chien, Orthogonal Latin square codes. IBM J. Res. Dev. **14** (4), 390–394 (1970)
52. A.R. Alameldeen, Z. Chishti, C. Wilkerson, W. Wu, S.-L. Lu, Adaptive cache design to enable reliable low-voltage operation. IEEE Trans. Comput. **60**(1), 50–63 (2011)
53. D. H. Yoon, M. Erez, Memory mapped ECC: low-cost error protection for last level caches, in *Proceeding of the 36th Annual International Symposium on Computer Architecture* (*ISCA*), 2009
54. P. Reviriego, C. Argyrides, J.A. Maestro, Efficient error detection in double error correction BCH codes for memory applications. Microelectron. Reliab. **52**(7), 1528–1530 (2012)
55. J. Kim, S. Kim, Y. Lee, SimTag: exploiting tag bits similarity to improve the reliability of the data caches, in *Proceeding Design Automation and Test in Europe*, 2010

56. S. Wang, J. Hu, S.G. Ziavras, Replicating tag entries for reliability enhancement in cache tag arrays. IEEE Trans. Very Large Scale Integr. Syst. **20**(4), 643–654 (2012)
57. P. Reviriego, S. Pontarelli, M. Ottavi, J.A. Maestro, FastTag: a technique to protect cache tags against soft errors. IEEE Trans. Device Mater. Reliab. **14**(3), 935–937 (2014)
58. P. Reviriego, S.S. Liu, A. Sánchez-Macián, L.Y. Xiao, J.A. Maestro, Unequal error protection codes derived from SEC-DED codes, IET Electron. Lett. (2016) (in press)
59. V. Sridharan, D. Liberty, A study of DRAM failures in the field, in *Proceeding of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2102
60. B. Schroeder, E. Pinheiro, W-D. Weber, DRAM errors in the wild: a large-scale field study, in *Proceeding of ACM SIGMETRICS*, 2009
61. H. Schmidt, M. Hermann, K. Grürmann, F. Gliem, V. Ferlet-Cavrois, Radiation hard memory. Radiation testing of candidate memory devices for Laplace mission, CNES/ESA Radiation effects final presentation days, March 2015
62. International Business Machines Corporation (IBM) "Chipkill Memory," http://www-05.ibm.com/hu/termekismertetok/xseries/dn/chipkill.pdf, Technical Report, 2012
63. X. Jian, H. Duwe, J. Sartori, V. Sridharan, R. Kumar, Low-power, low-storage-overhead chipkill correct via multi-line error correction, in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis* (*SC*), 2013
64. S. Pontarelli, G.C. Cardarilli, M. Re, A. Salsano, Error correction codes for SEU and SEFI tolerant memory systems, in *24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, October 2009, pp. 425–430
65. D.H. Yoon, M. Erez, Virtualized and flexible ECC for main memory, in *Proceeding of the International Symposium on Architectural Support for Programming Languages and Operating Systems*, 2010
66. A.N. Udipi, N. Muralimanohar, R. Balsubramonian, A. Davis, N.P. Jouppi, LOT-ECC: localized and tiered reliability mechanisms for commodity memory systems, in *Proceeding of the International Symposium on Computer Architecture*, 2012
67. J. Kim, M. Sullivan, M. Erez, Bamboo ECC: strong, safe, and flexible codes for reliable computer memory, in *Proceeding of the International Symposium on High Performance Computer Architecture*, 2015
68. C. Weis, I. Loi, L. Benini, N. Wehn, Exploration and optimization of 3-D integrated DRAM subsystems, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.**32**(4) (2013)
69. A. Schoenberger, K. Hofmann, Analysis of asymmetric 3D DRAM architecture in combination with L2 cache size reduction, in *Proceeding of the IEEE High Performance Computing & Simulation* (*HPCS*), 2015
70. R Kumar, V Zyuban, D.M. Tullsen, Interconnections in multi-core architectures: understanding mechanisms, overheads and scaling, in *32nd International Symposium on Computer Architecture* (*ISCA'05*), 2005
71. D. Bertozzi, L. Benini, G. De Micheli, Error control schemes for on-chip communication links: the energy-reliability tradeoff. IEEE Trans. Comput.-Aided Des. of Integr. Circuits and Syst. **24**(6), 818–831 (2005)
72. L. Benini, G. De Micheli, Networks on chips: a new SoC paradigm. IEEE Comput. **35**(1), 70–78 (2002)
73. S. Murali, N. Vijaykrishnan, M.J. Irwin, L. Benini, G. De Micheli, Analysis of error recovery schemes for networks on chips. IEEE Des. Test Comput. **22**(5), 434–442 (2005)
74. A. Ejlali, et al., Joint consideration of fault-tolerance, energy-efficiency and performance in on-chip networks, in *Proceeding of Design, Automation and Test in Europe Conference and Exhibition*, 2007
75. B. Fu, P. Ampadu, Error control for network-on-chip links (Springer Science & Business Media, 2011)
76. S. Lee et al., Low-power, resilient interconnection with orthogonal Latin squares. IEEE Des. Test Comput. **28**(2), 30–39 (2011)

77. D. Rossi, C. Metra, K.A. Nieuwland, A. Katoch, Exploiting ECC redundancy to minimize crosstalk impact. IEEE Des. & Test Comput. **22**, 59–70 (2005)
78. A. Ganguly, P.P. Pande, B. Belzer, Crosstalk-aware channel coding schemes for energy efficient and reliable NOC interconnects. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **17**(11), 1626–1639, 2009
79. S. Sridhara, R.N. Shanbhag, Coding for reliable on-chip buses: a class of fundamental bounds and practical codes. IEEE Trans. Comput. Aided Des. Integr. Circuits and Syst. **5**, 977–982 (2007)
80. T. Lehtonen, P. Lijieberg J. Plosila, Analysis of forward error correction methods for nanoscale networks-on-chip, in *Proceedings of the nano-net, 2007*, Catania, Italy, pp. 1–5
81. B. Fu, P. Ampadu, On hamming product codes with type-II hybrid ARQ for on-chip interconnects. IEEE Trans. Circuits Syst. I, Regul. Pap. **56**(9), 2042–2054 (2009)
82. P. Ampadu, Q. Yu, B. Fu, Reliable networks-on-chip design for sustainable computing systems, in *Design Technologies for Green and Sustainable Computing Systems* (Springer New York), pp. 23–57
83. C. Feng, Z. Lu, A. Jantsch, M. Zhang, Z. Xing, Addressing transient and permanent faults in NoC with efficient fault-tolerant de-flection router. IEEE Trans. Very Large Scale Integr. VLSI Syst. **21**(6), 1053–1066 (2013)
84. S. Dumitras, R. Kerner, R. Marculescu, Towards on-chip fault-tolerant communication, in *Proceedings of the Asia and South Pacific Design Automation Conference* (*ASP-DAC'03*), Kitakyushu, Japan, pp. 225–232
85. Z.J. Haas, J. Y. Halpern, L. Li Gossip-based ad hoc routing. IEEE/ACM Trans. Networking **14**, 476–49, 2006
86. M. Pirretti et al., Fault tolerant algorithms for network-on-chip interconnect, in *Proceeding IEEE Computer Society Annual Symposium on VLSI Emerging Trends in VLSI System Design,* (*ISVLSI'04*), Lafayette, Louisiana, USA, 2004, pp. 46–51
87. T. Lehtonen, D. Wolpert, P. Liljeberg, J. Plosila, P. Ampadu, Self-adaptive system for addressing permanent errors in on-chip interconnects. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **18**(4), 527–540 (2010)
88. C. Grecu, P. Pande, A. Ivanov, R. Saleh, BIST for network-on-chip interconnect infrastructures, in *Proceedings of the 24th IEEE VLSI Test Symposium*, 2006
89. M. Radetzki, C Feng, X Zhao, A Jantsch, Methods for fault tolerance in networks-on-chip *ACM Comput. Surv.* **46**(1), pp. 8:1, 8:38 (2013)
90. E. Bolotin,I. Cidon, R. Ginosar, A. Kolodny, "Routing table minimization for irregular mesh NoCs", In proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'07). 1–6
91. J. Flich, A. Mejia, P. Lopez, J. Duato, Region-based routing: an efficient routing mechanism to tackle unreliable hardware in network on chips, in *Proceedings of the Symposium on Networks-on-Chip,* (*NOCS'07*), 2007, pp. 183–194
92. Z. Zhang, A. Greiner, S. Taktak, A reconfigurable routing algorithm for a fault-tolerant 2D-mesh network-on-chip, in *Proceedings of IEEE Design Automation Conference* (*DAC'08*), 2008, pp. 441–446
93. C. Bobda et al., DyNoC: a dynamic infrastructure for communication in dynamically reconfigurable devices, in *Proceedings of International Conference on Field Programmable Logic and Applications,* (*FPL08*), 2008, pp. 153–158
94. M. Valinataj, S. Mohammadi, J. Plosila, P. Liljeberg, A fault-tolerant and congestion-aware routing algorithm for Networks-on-chip. DDECS **2010**, 139–144 (2010)
95. M. Ebrahimi, M. Daneshtalab, J. Plosila, H. Tenhunen, MAFA: adaptive fault-tolerant routing algorithm for networks-on-chip. DSD **2012**, 201–207 (2012)
96. M. Dimopoulos, et al., Fault-tolerant adaptive routing under permanent and temporary failures for many-core systems-on-chip, in *Proceeding of the 9th IEEE International On-Line Testing Symposium* (*IOLTS13*), 2013
97. W.J. Dally, C.L. Seitz, Deadlock-free message routing in multiprocessor interconnection networks. IEEE Trans. Comput. **36**(5), 547–553 (1987)

98. K. Constantinides et al.. Bulletproof: a defect-tolerant CMP switch architecture, *in Proceeding of the 12th IEEE International Symposium on High-Performance Computer Architecture*, 2006, pp. 5–16
99. D. Fick, A DeOrio, J. Hu, V. Bertacco, D. Blaauw, D. Sylvester, Vicis: a reliable network for unreliable silicon, in *Proceedings of the 46th ACM Annual Design Automation Conference,* (*DAC'09*), 2009, 812–817
100. A. Kohler, M. Radetzki, Fault-tolerant architecture and deflection routing for degradable NoC switches, in *Proceedings of the 3rd ACM/IEEE International Symposium on Networks-on-Chips,* (*NOCS'09*), 2009, pp. 22–31

Application-Specific Solutions

Viacheslav Izosimov, Antonis Paschalis, Pedro Reviriego and Hans Manhaeve

Abstract This chapter discusses surface transportation applications, space applications, and medical applications in detail. It extends the discussion from Chap. 3 where we considered a broader variety of application domains and their relation to dependability. The choice of these applications is due to expertise of the authors and positioning of these applications in the overall dependability palette as ones of the most challenging yet different from each other.

1 Automotive and Transport Applications

1.1 Introduction

In this section, we introduce automotive and transport systems, in passenger cars, buses, commercial vehicles, and transport applications such as railroad systems. Automotive is possibly one of the most challenging application domains for electronic systems in transportation with tough cost constraints due to mass-market manufacturing, tough competition between players and a great level of innovations with new technologies coming onto the market with introduction of self-driving and electric vehicles. Hence, a lot of emphasis in this section will be on automotive

V. Izosimov (✉)
Semcon Sweden AB, Linköping, Sweden
e-mail: viacheslav.izosimov@semcon.com

V. Izosimov
KTH Royal Institute of Technology, Stockholm, Sweden

A. Paschalis
University of Athens, Athens, Greece

P. Reviriego
Universidad Antonio de Nebrija, Madrid, Spain

H. Manhaeve
Ridgetop Europe, Bruges, Belgium

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_6

vehicles. At the same time, we will look into related transportation applications wherever appropriate.

1.2 Domain-Specific Manufacturability- and Reliability-Related Challenges

In today's transportation systems, thousands of microelectronic chips are used for power train, safety, comfort, and infotainment applications and the embedded software content is measured by hundreds of megabytes. On top, the trend toward electric mobility poses further challenges in today's architectures for transportation systems. And while today's robustness level of classical cars has been accomplished over 125 years, at least the same level has to be accomplished within 10–15 years for hybrid and fully electric vehicles. In effect, the complexity of any electronics in a vehicle directly drives the robustness requirements. Doubling the electronics content simply calls for cutting failure rates of electronic components in half, to maintain the overall quality level. Furthermore, when newer semiconductor technologies were developed in the past they would go through a 5-year maturing period in the consumer sector before being introduced to automotive electronics. In more recent times, however, the increasing demand for higher performance devices, developed under much tighter cost pressures, means that there is now a shortening of the maturing time to just one year. In Fig. 1, we present a generalized example of

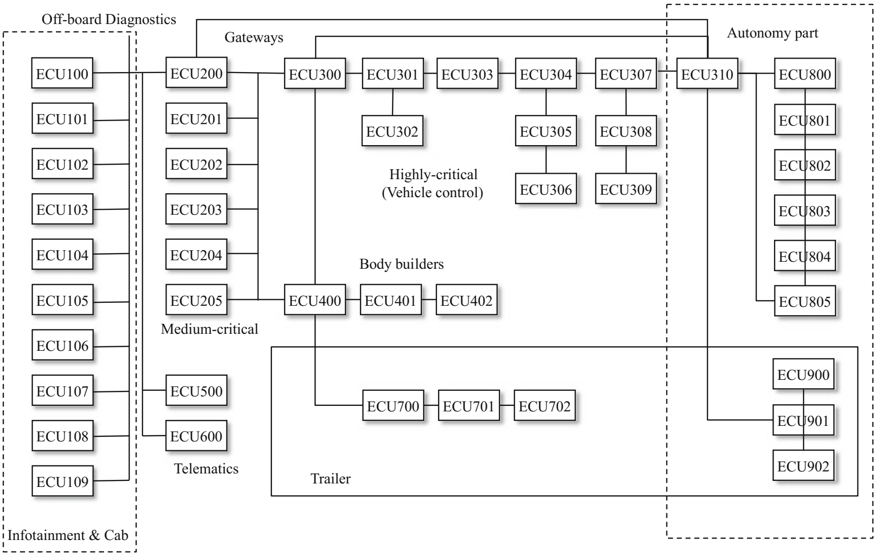


Fig. 1 A Generalized E/E Architecture of Autonomy-Ready Commercial Vehicle (not linked to any particular vehicle manufacturer)

a modern E/E (Electrical/Electronic) architecture of an autonomy-ready commercial vehicle. On the left-hand side, the infotainment cluster is shown and, on the right-hand side, we outline the autonomy add-on with autonomous driver (AD) ECU310 and its perception functionality. ECU310 has direct access to the main vehicle driving functionality (ECU300-309), perception modules on the truck (ECU800-805) and on the trailer (ECU900-902). Via gateway ECU300, it accesses body builders (ECU400-402), trailer control (ECU700-702) and other units (ECU200-205). Via gateway ECU200, it connects to infotainment (ECU100-109), telematics (ECU500 and ECU600) and off-board diagnostics.

The number of units and configurations in the architecture can be changed to fulfill particular application needs, from very simple configurations with only few computation units for short-range lorries to the most complex long haulers. The E/E architecture can be instantiated to variety of vehicles and configured accordingly. The architecture in Fig. 1, for example, can be instantiated to a whole spectrum of commercial vehicles and buses, both autonomous and non-autonomous, from 2 to 6 axles, with hybrid and traditional drivetrains. This approach provides advantages for mass-market yet enabling variability and flexibility requested by customers. However, at the same time, it creates a great number of challenges in dependability, safety and security. For example, complete verification of all software interactions in each vehicle variant may not be even possible. Threat analysis of security vulnerabilities is becoming a troublesome and time-consuming task.

Thus, with the introduction of advanced driver assistance systems such as X-by-wire applications, (semi-)autonomous driving and platooning, fault tolerance is gaining a momentum in automotive industry, which considers complex algorithms, e.g. for environment sensing, situation perception and reasoning in a closed-loop control systems. It is no longer sufficient or can be even dangerous to just switch off a system component. Loosing of, for example, steering at high speeds will lead to an uncontrollable vehicle, which can be crashed with severe injuries before the driver can stop the vehicle with braking. A certain level of functionality must remain until the vehicle can be fully and safely stopped. Similarly, a certain driving functionality must be present in the trains and trams, where the functionality of brake system must be present until the full stop. However, in case of trains, a full stop is no longer a safe state and a certain level of functionality must remain even a bit further, to move the train out of a tunnel or from a bridge at a low speed.

1.3 Current Practice

1.3.1 Architecture-Driven Dependability Approaches

One of the emerging trends in transportation industry is to merge several applications with different criticalities into less number of electronic control units (ECUs). This can be often implemented on a multi-core chip, where each core handles applications with the same level of criticality. On this chip, isolation between different criticality applications must be provided even if executed on different cores, with respect to the logical area, communication buses, memory and I/Os. Moreover, cores must have individual power supplies such that a failed core can be safety switched off and the applications can be safely moved to a backup core. Fault tolerant actions on one core must be transparent, e.g. must not affect executions on the other cores nor timing of other executions.

The current state of the art, however, is that ECUs are still "one core." Although the dual-core hardware is often used, the redundancy is usually used for error detection purposes, for example, for Lockstep [1]. The separation between different criticality applications is usually logical (e.g., in software), where a "safe switched context" [2] is responsible for switching between the applications. No hardware support for this functionality is usually provided. Moreover, no fault tolerance action is activated for a failed application process; instead a reboot is triggered by an external watchdog. The reason for this implementation is simplicity. It has been known that fault tolerance introduces complexity and unnecessary complexity is a source of issues and can lead to delays in development. In addition, it is often possible to reboot the ECU within the given "fault tolerance" interval (time until a fault effect becomes safety-critical) and restore its functionality. As the sources of faults are not known, reboot is the safest way to overcome them.

However, these solutions become increasingly limited with the amount of applications growing. Moreover, it could be possible to switch off or "reboot" some of the applications but not all. Reboot is not a good solution in case of multiple faults. The fault may simply re-appear after the reboot and the new reboot would be needed. Thus, the system may never finish rebooting or it will exceed the "fault tolerance" interval crashing the vehicle. The risk of multiple faults increases with the increased levels of integration in semiconductors, low power, aging effects, variability, increased frequency, and often hazardous environments with tough temperature profiles, emissions and vibrations. Hence, other approaches for fault tolerance than reboot are necessary. However, it is essential to overcome "complexity" of more fine-tuned fault tolerance solutions such as reexecution and rollback recovery [3] and introduction of truly multi-core platforms with clear isolation between applications possible.

In Fig. 2, we present example of a redundant (diversified) steering system that is enabled with electrical power train combined with the traditional steering (the ASIL D refers to the highest criticality level). This type of diversification is necessary for highly critical applications such as self-driving. For example, the electric ASIL D All-Wheel-Drive (AWD) with the torque vectoring (TV) functionality can be used for a second "channel" of steering because it enables efficient changing of the direction of the vehicle and is de facto independent from the traditional steering (unless a common control channel is used) [4].

It is also becoming more demanding to enable fail-operational behavior of the electronic units versus traditional fail stop. (We will further elaborate on the requirements for future applications in Sect. 1.4). In case the fail operation behavior should be enabled, a possible solution in automotive is illustrated in Fig. 3. Note

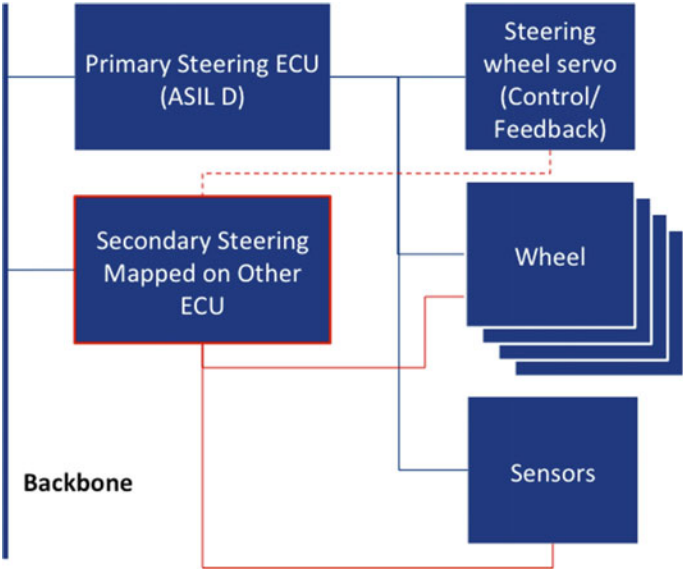

Primary Steering ECU (ASIL D)
Steering wheel servo (Control/ Feedback)
Secondary Steering Mapped on Other ECU
Wheel
Sensors
Backbone

Fig. 2 Fail-operational steering (ready for automated driving)

that we provide this visualization for illustrative purpose (to show the current state-of-praxis in automotive for frontier applications such as self-driving vehicles). The implementation details are changed and/or omitted compared to the actual system and, as always, different applications may require different solutions and each system should be analyzed in the given application context. (For example, a common misunderstanding in automotive is that the watchdog-based solutions are always good, which is not true and may, in fact, even be dangerous.)

Let us now explain some of the technical terms considered in this example. The goal is to construct an ASIL D fail-operational platform, which means that the reliability target is about 10 FIT (10^{-8} failures per hour) for dangerous failures.

First of all, automotive industry (to a large degree) uses standardized platforms called AUTOSAR (with the version 4.2 presently the newest one for commercial use) [5]. The AUTOSAR is a (large) set of requirements targeting inter-operability and reuse of automotive software and hardware, with particular focus on software-based solutions. From version 4.0 (also considering version 3.2), AUTOSAR is aligned with the ISO 26262 automotive functional safety standard [6] (e.g. safety analysis is conducted on AUTOSAR modules and adjustments such as Safety Watchdog, Safe Switching Context and End-to-end protection introduced [2]). In Fig. 3, we consider, in particular, End-to-end protection for ensuring "safe" communications between the modules. Each module in itself should implement at least Safety Watchdog (implemented in software) connected to the External Hardware Watchdog (on a dedicated pin). In Fig. 3, the main functionality is redundant, namely MCU1 and MCU2 run in parallel. Failure of any MCU will trigger external watchdog followed by reset of that MCU. The other MCU will continue to operate.

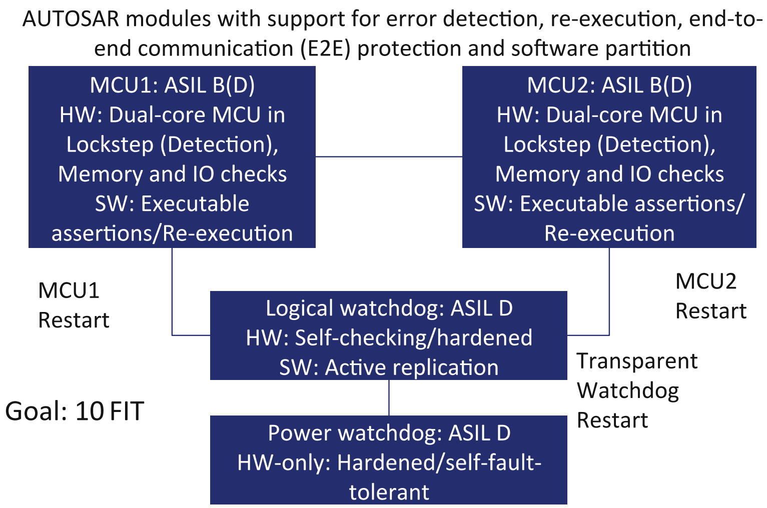
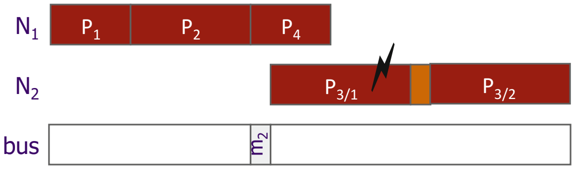

AUTOSAR modules with support for error detection, re-execution, end-to-end communication (E2E) protection and software partition
MCU1: ASIL B(D)
HW: Dual-core MCU in Lockstep (Detection), Memory and IO checks
SW: Executable assertions/Re-execution
MCU2: ASIL B(D)
HW: Dual-core MCU in Lockstep (Detection), Memory and IO checks
SW: Executable assertions/ Re-execution
MCU1 Restart
MCU2 Restart
Logical watchdog: ASIL D
HW: Self-checking/hardened
SW: Active replication
Transparent Watchdog Restart
Goal: 10 FIT
Power watchdog: ASIL D
HW-only: Hardened/self-fault-tolerant

Fig. 3 Fail-operational ECU (ready for automated driving)

Note that these two redundant MCUs implement so-called *decomposition* (according to ISO 26262 terminology [6], see Part 9). The original ASIL D operational requirements (of the highest criticality) are decomposed into ASIL B requirements (of lower criticality). ASIL D software is often considered very costly and, hence, lowering of the requirements to ASIL B even despite of the redundancy is a more attractive design option. Note also that software partitioning is considered in software to facilitate protection between software processes on the operating system level (a technique that separate processes from each other, not allowing direct communication, and monitoring behavior of the processes). Software partitioning can be implemented with *virtualization*, *hypervisor* techniques (bare metal and native) [7] or, recently proposed, *container* techniques. The latter one is known for its lightweight character and finds more and more applications. The most interesting implementation of containers is *Docker* [8], originating from the Linux community.

To facilitate soft error detection, each MCU implements presently common (and relatively low cost) lock-step functionality (two cores run in parallel with time shift and continuously checking each other) and Executable Assertions (or *observations points*) in software connected to the AUTOSAR Safety Watchdog. Assertions are necessary for functioning of the Safety Watchdog (that itself can only collect information from the observation points) and, at the same time, one of the most efficient software-based error detection techniques. It is also possible to "re-use" assertions from the development phase, used for debugging purposes, and reintroduce them into the actual production code [9]. As of it is now, each fault either from assertion (through the Safety Watchdog) or from the Lockstep (connected directly to the External Watchdog) will trigger restart of MCU. Restart is costly and may be inefficient especially in case of multiple faults (albeit is still the most common practice in automotive). Hence, in this case, the suggestion is to use the

Fig. 4 Re-execution example



next simplest fault tolerance solution after restart—re-execution [3]. Re-execution is depicted in Fig. 4. In the example, that contains 4 processes run on two nodes, process P_3 is reexecuted after fault. Note the re-execution overhead between two process executions, which is needed to restore processor's state after the failed process. In this case, the restart is not necessary still the solution is rather simple. Moreover, re-execution is supported in many operating systems and fault tolerance scheduling code for re-execution is rather simple. In case of re-execution, each process has to also contain error detection (e.g., executable assertions) which overhead we have included into the process execution time in Fig. 4.

Another alternative to re-execution is active replication [3], which is used for the Logical Watchdog in Fig. 3. In case of active replication (implemented in software), replicas of processes are always executed and that process that is not faulty is allowed to send outputs, while the faulty one will silently terminate. This is possible due to error detection mechanisms. In case of the Logical Watchdog unit, error detection (self-checking) is implemented in hardware with the technique called *hardening* [10]. Hardware-based error detection techniques are efficient in terms of execution time but require extra hardware, often specialized. Since the Logical Watchdog has to be implemented to the (highest) ASIL D level and since the software of the watchdog is rather simple, hardening can be motivated in this case.

Finally, the power watchdog is necessary in automotive due to significant implications of the power quality on the electronics and often variable quality of power supply. Hence, it is often necessary to have more than one power source. For example, 24 V power source can be complemented with 48 V power source or 600 V power source (supplied after DC/DC or AC/DC power conversions). The power watchdog does not have to have software at all but is required to be self-fault tolerant (with hardware-based fault tolerance for error correction).

The last two points to consider in the design of fail operational electronic control unit (ECU) are memory protection and I/O (integrity) checks, that are necessary to ensure proper quality of data stored in the memory and absence of insufficient/incorrect/(or even malicious) input data. A number of techniques (in combination) are often used for memory protection, with memory management units (MMUs), error detection and error correction codes (ECC), redundant memory controllers, and alike [11]. Periodic integrity checks (against *sleeping* faults, also called dormant or *latent*) are often demanded for memories. The integrity checks of I/O involve out-of-norm assertions (including electrical level), signal monitoring, and authentication and authorization checks. Although security is not considered extensively in this book, it is becoming an emerging topic, in particular, for automotive as the number of vehicles hacked were reported. The problem will become even more apparent for highly automated vehicles. Examples of methods to deal with this type of errors can be found in [12].

1.3.2 Technology-Driven Dependability Approaches

The state of the art in semiconductor qualification in the automotive industry, also including commercial vehicles and buses, is based on the AEC-Q100/AEC-Q200 standard [13], which has been defined by the major automotive players. It defines the minimum stress test-driven qualification requirements. After fulfilling these requirements in the standard, the device can be expected to give a certain level of quality/reliability in the application. However, the robustness validation working group has shown already in 2006–2007 that this process describes only the minimal stress test-driven qualification requirements and does not apply to describe the triggering of complex failure mechanisms in today's semiconductors [14]. The working group has proposed a mission profile-driven robustness validation process to solve that problem. A mission profile defines the condition of use for the semiconductor component in the intended application. Robustness validation is a knowledge-based approach that uses stress tests that are defined to address specific failure mechanisms using suitable test vehicles and stress conditions on the physical device [14]. Many automotive manufacturers have also internal standards, which exceed the basic standards. Moreover, ISO 26262 in part 8 "Supporting Processes" [6] provides additional guidelines on hardware components qualification, where such aspects as context and statistical information can be used to further enhance qualification of the hardware components. Further, in the upcoming update of ISO 26262, Part 11 introduces guidelines on using semiconductor technology. Another part of ISO 26262, part 5 on hardware development [6], includes a lot of important guidelines on development of hardware components. In particular, it includes target fault rates and categorization of fault types for meeting safety requirements, which should be used for assessment of whether reliability requirements on hardware components have been met.

In order to handle the strongly reduced maturing time in future, this robustness validation process has to be closely integrated within semiconductor design flow in order to optimize the chip for its final application by covering the entire supply chain. OEMs have usually a clear idea on how electronics is applied. For example, power switches used to fire an airbag may be switched only a very small number of times. Power switches in a bridge to automatically shift gear—especially if used in pulse-width modulation—may be switched millions of times. Many more pieces of information, e.g., temperature, voltage, and current profiles, may be of interest. All this information has to be collected in a systematic way, thus creating a standardized mission profile [15].

With commercial vehicles and buses, a number of additional international and national regulations would apply due to an increased level of severity, for example, for brakes and emergency brakes. Furthermore, the upcoming update of ISO 26262 will apply for trucks and buses. When talking about transportation, we should not forget important requirements for railroad vehicles and their hardware components. Railroad vehicles are considered as SIL 4 (multiple casualties possible) and another set of standards would apply, namely CENELIC EN 50126/128/129 [16–18]. Stress conditions and individual installations shall be considered in railroad and approved through the respective certification bodies. For example, in Germany, each train must undergo separate certification process and must have a dedicated set of safety documentation.

Designing for robustness in the transportation context means to translate system hazards (e.g., potential dangerous situations) into the electronics requirements and then into constraints, which can be maintained and checked throughout comprehensive and precise verification and validation process for microelectronic products. This process has to work for analog and digital constraints. In this way, the requirements compliance and the related robustness have to be "built-in" in the design instead of post qualification of hardware components. Both lifetime requirements and environmental conditions must be considered in the design.

Lifetime requirements can include the following:

- Expected operating lifetime of the device in power-on hours
- Actual number of weekly operating hours
- Device operating voltage and electric field
- Number of mini-cycles and sleep cycles
- Early-life/cumulative End-of-Life failure rate.

The following environmental conditions are often of interest:

- Number of environmental and power cycles experienced per day
- Ambient relative humidity and temperature range of the environment
- Electromagnetic and Electrostatic Fields
- Vibrations and mechanical stress
- Voltage/Glitch overload
- Other relevant environmental conditions (e.g., close proximity to cooling channels, interaction with liquids, chemical substances, etc.).

Note also that these conditions and lifetime requirements are enhanced with introduction of highly automated vehicles where, in particular, environment perception system is considered as the most critical bottleneck. It means that such environmental conditions as sunlight intensity, rainy/snowy weather, external (with respect to vehicle) electromagnetic fields play role. Even mechanical conditions of the windshield and physical (dis)-placement of sensors become safety-critical.

1.4 Future of the Domain

In transportation systems, robustness has been always included as an essential design attribute. However, with the increased complexity of systems, increased requirements on reliability of hardware components and tightened requirements to safety and quality, considering of fault tolerance in system design is encouraging and, in some applications, such as autonomous driving is demanding. The number of challenges should be resolved before the fine-tuned fault tolerance can be applied in practice in the transportation domain in addition to all the effort for increasing of system dependability with the mission profile-driven approach:

1. Systematic reliability modeling of applications, platforms and fault tolerance actions for different operational profiles and environments, with the software tool support.
2. "Proven" fault tolerance architectural patterns and "standardized" components for multi-core systems. This can be achieved through standardization effort in the AUTOSAR community of automotive industry and the respective efforts in railroad.
3. Low-cost mass-produced multi-core platforms suitable for mixed criticality systems. For example, the recent development of Infineon's AURIX platform is an interesting example.
4. Software tool support for enabling and verification of fine-tuned fault tolerant solutions. For example, it can be included as part of the AUTOSAR configuration tools and the tools provided by hardware suppliers with the fault tolerant hardware components.
5. Software tool support for evaluation of whether isolation between applications with different criticalities has been achieved and for indication of which steps must be performed to provide the isolation. This can be included into system modeling and verification and validation tools.
6. A classic fault tolerance meets with algorithms and safety-critical sensors in the environment perception block of self-driving vehicles, which is, by far, the most challenging problem that is faced by automotive industry today, with barely any solution available. Still, in many current "autonomous" vehicles, such as Tesla models and Mercedes E-class, the full responsibility is placed on the driver. In many recent accidents with Tesla vehicles, for example, the driver is often the one to blame who over-trusted the system. Knowledge in advanced mathematics and the dependability knowledge should meet to enable future dependable self-driving vehicles, trustworthy enough to be driven by general public. The DriveMe project with 100 autonomous Volvo Cars vehicles driving around Goteborg city in Sweden [in 2017–2018] is, hence, an interesting case to observe. There a car will take a greater responsibility.
7. Relation towards connectivity is yet another challenging problem. It is not possible to set dependability requirements on "air" and external actors such as computing clouds, while the decisions, not least safety-critical, have to be often based on this external information. Moreover, security issues further complicate the picture with malicious actors increasingly interested in assets of the vehicles alone or as the whole fleet. The problem is an increasing concern for Intelligent Transportation Systems (ITS) and, at present, research there is still very limited and ad hoc solutions dominate practical implementations.

2 Space Applications

2.1 Introduction

Space is another challenging application domain for electronic systems, to some degree even more challenging than automotive, not least due to tougher environment. Circuits that operate in space are subject to extreme environmental conditions in terms of temperature and radiation. They also suffer mechanical stress during launching. Component replacement or maintenance operations are difficult or impossible. This means that space-grade devices require special care during their design, manufacturing, and qualification processes. One of the key challenges is how to mitigate the effects of radiation on electronic circuits [19].

Radiation sources are multiple: some take their origin in the Sun (e.g. solar flares, coronal mass ejection and solar wind) and others come from outside the solar system (galactic cosmic rays). Electromagnetic radiation (through the entire spectrum, from radio waves to X-rays and gamma rays), as well as, ionizing radiation of accelerating particles (mostly protons and electrons, but also heavy ions) are emitted from the Sun. Galactic cosmic rays are high-energy charged particles coming from outside the solar system and generally from within the Milky Way galaxy. They are composed of about 89% of protons, 10% of Helium nuclei, the remaining 1% being fully ionized nuclei of heavier elements and electrons. Besides, there are two regions of the earth magnetosphere, the Van Allen Belts, where high-energy particles, mainly protons and electrons, are trapped by the Earth's magnetic field. Mostly high-energy protons can create neutrons, secondary protons, muons and neutrinos by spallation reaction on atmospheric nuclei. Neutrons cannot cause errors in semiconductor devices through direct ionization, as it is the case with protons and heavy ions, but they can generate errors through nuclear reactions with silicon resulting in recoils which may deposit enough charge in a small volume to trigger an error event. The importance of each radiation type and source depends on the orbit of the space mission and for all cases radiation is much larger than the one considered for terrestrial applications.

The radiation effects in semiconductor devices can be classified as: (a) total ionizing dose (TID); (b) displacement damage (DD), and (c) single event effects (SEEs).

Total ionizing dose (TID) degradation due to buildup of charge in insulating layers, and it has a cumulative effect on electronics, resulting in a gradual loss of performance and eventual failure. Displacement damage (DD) effect which results from damage to the crystalline structure of semiconductors due to particles losing energy, not by way of ionization, but by elastic/inelastic collisions with nuclei in the target material (non-ionizing dose effect). Single event effects (SEEs) arise from the interaction of single particles (e.g., protons, neutrons or heavy ions) with the semiconductor causing either destructive permanent effects or transient effects in memories and registers. The destructive permanent effects include single event latch-ups (SEL) in CMOS circuits, single event snapbacks (SESB) in NMOS devices, single event gate/dielectric ruptures (SEGR/SEDR) and single event burnouts (SEB) in power transistors. The transient effects includes single event upsets (SEU) that produces bit-flips leading to change of stored information, multiple-cell upsets (MCU) in memories and registers including single word multiple adjacent bit upsets (MBU), single event functional interrupt (SEFI) in control circuitry and single event transient (SET) in linear circuits, i.e., a current transient interpreted as a false signal [19, 20].

Dependability [21] is a major issue in space applications for the following reasons:

(a) The space environment is harsh for space electronics with respect to radiation and the amount and types of radiation present in most space missions are much larger than those experienced in terrestrial applications.
(b) The cost of a failure in a space mission can be catastrophic with respect to safety [22] leading to loss of life, life threatening or permanently disabling injury or occupational illness, loss of an interfacing manned flight system, severe detrimental environmental effects, loss of launch site facilities, and loss of system causing major disruptions to scientific or commercial plans.
(c) The nature of space applications makes maintenance hard and in many cases impossible and thus dependability is critical to ensure system reliability, availability and maintainability for the duration of the mission.

Mitigation of radiation effects is not only mandatory for space applications, but it is becoming mandatory for an increasing number of application domains, including networking, servers, avionics, medical, and automotive electronics due to drastic semiconductor device shrinking, very low operating voltages, increasing complexities, and high clock frequencies at ground level. The most radiation challenging space mission is the 10-year NASA mission to the Europa moon of Jupiter, where semiconductor devices able to withstand 2.9 Mrad TID are required. This level is more than twice that required for U.S. defense systems to be operated through nuclear explosions, and 7 times greater than any previous NASA mission.

The main computing systems used in space applications are known as Payload Data Processing Units (PDPUs) and are used as an interface between the spacecraft and several payload instruments, providing control of payload systems, processing of telecommands and processing of acquired science data. The space environment presents special challenges to the PDPU system designer. The major among these are:

(a) Power efficiency since electrical power is a scarce and expensive commodity for orbiters and deep-space probes.
(b) Minimum size and weight since increased size and weight increase launch costs and require more fuel for on-orbit maneuvering.
(c) Adequate reliability at moderate unit costs.
(d) Adaptability to meet new or changing mission requirements.
(e) Mission-specific radiation tolerance.

Control, monitoring, and telecommand processing tasks require low data rates, thus can be efficiently implemented in software by a general purpose processor. The huge amounts of data generated from today's and future high resolution and high-speed imagers and image spectrometers in combination with the limited downlink bandwidth requires high-performance on-board processing to handle high data rates and volumes that can be achieved only by dedicated hardware [23].

The required on-board processing of space mission payload data is a challenging task for spacecraft data processing units, since data rates and data volumes produced by payloads continue to increase, while the available downlink bandwidth to ground stations is comparatively stable. The established space-qualified PDPUs are now outdated as much higher performance is already needed.

The new space-qualified requirements of PDPUs for future space missions, as described in the dedicated ESA round table synthesis report [24], can be summarized as:

(1) High processing speed (from 10–100 MIPS/MFLOPS for science and robotic exploration missions (e.g., EUCLID, PLATO) to 10,000 MIPS/MFLOPS for earth-observation missions).
(2) Low power consumption (few Watts per 1000 MIPS/MFLOPS).
(3) Radiation hardness (100–1000 Krad TID).
(4) Extremely high reliability, protected memories (error detection and correction mechanism) and other key-functional modules.
(5) High-quality software development.
(6) Support of space standard interfaces.

The rest of this section presents first the current solutions used to ensure that processing systems in space applications are reliable and then it discusses the research efforts that are leading to the next generation of processing systems for space.

2.2 Current Solutions

Significant efforts have been made during the past years to cope with the undesired effects induced by radiation. As result, a wide scope of techniques has been adopted to mitigate these effects at the different abstraction levels of the microelectronics development flow: manufacturing process level, layout level, architecture and design level, system and software level [25].

At process level are applied techniques concerning the manufacturing processes, also known as radiation hardening by process (RHBP). These techniques mitigate TID and SEE, and generally concern modifications of doping profiles in devices and substrates, optimization of deposition processes for insulators and use of specific materials (e.g., epitaxial layers, buried layers, silicon on insulator, silicon on sapphire, triple well technology, dry thermal oxidation, implantation of Al, Si, P, Fl, and As elements into oxides).

At layout level are applied techniques aiming at optimizing transistor's layout and placement and inserting protection elements in order to reduce mainly TID and SEL phenomena (e.g., enclosed layout transistor, contacts, and guard rings). Radiation-hardened libraries for ASIC design have been developed based on commercial CMOS technologies, like the IMEC design against radiation effects (DARE) library at 180 nm, the CERN library at 240 nm, the BAE library at 150 nm, the Ramon libraries at 180 and 130 nm, the Aeroflex libraries at 600, 250, 130, and 90 nm, the ATMEL MH1RT at 350 nm and ATC18RHA at 180 nm, the ATK at 350 nm, and ST Microelectronics at 65 nm.

At architecture and design level are applied techniques specific to the circuit's nature (digital, analog or mixed signal) and/or to the circuit's family (application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs) or embedded memories). The majority of them belong to main approaches such as redundancy (hardware or temporal) or error detection and correction (EDAC). In ASICs, hardware redundancy is based on replicating sensitive resources and voting the outputs to detect discrepancies (duplex architectures for SEU detection and triple modular redundancy (TMR) for SEU correction), while temporal redundancy is based on sampling data at different instants and eliminates SET in combinational logic. Radiation-hardened ASIC processors (mainly RISC) and computers have been developed by several manufacturers for several NASA and ESA missions (e.g., the dual-core LEON3-FT SPARC V8 Processor chip (GR712RC) from Aeroflex Gaisler).

Embedded memory cells become radiation hardened by applying cell layout optimization imposing area and power overheads (extra resistors and capacitances to increase bit-flip threshold, increase of transistor size, and increase of the number of transistors of the cell based on two fundamental concepts: redundant storage of the information and feedback paths in order to restore the correct data). Representative hardened cells are: IBM and NASA-Whitaker with 16 transistors, NASA-Liu with 14 transistors, and HIT and DICE with 12 transistors. Apart from this, EDAC schemes are widely used for memory protection and are based on information redundancy, i.e., the stored information has some extra parity bits that form an error detection and correction code (such as parity and SEC-DED codes). EDAC schemes are combined with scrambling (the logic structure differs from the physical structure) in order to mitigate not detectable single word multiple adjacent bit upsets (MBU).

FPGAs are composed of two "layers": an operative layer containing the user logic and memory and a configuration layer determining the functionality of the user logic. The nature of the configuration layer depends on the type of FPGA: The antifuse FPGAs are one-time programming (OTP) and the configuration layer is immune to bit-flips provoked by radiation. The SRAM-based or flash-based FPGAs consist of memory cells that offer the advantage to be reconfigurable making possible "on-line" configuration of the FPGAs. According to the memory cell technology, it can be more or less sensitive to radiation. Indeed, bit-flips occurring in the configuration memory may have an impact on the application behavior in case that the perturbed bit is used. In such a case, FPGA memory reconfiguration is required to recover the nominal configuration. In FPGAs are applied techniques like local and global triple modular redundancy (TMR) with specific voter insertion, embedded user memory (BRAM) and configuration memory error detection and correction encoding schemes, reliability-oriented place and route algorithm (RoRA) to optimize the place and route process in the design flow, temporal redundancy and full or partial configuration memory scrubbing for SRAM-based FPGAs (periodically reload of the bit stream without operation disrupting). Radiation-tolerant or hardened FPGAs are provided by XILINX, Microsemi (Atmel), Aeroflex and Atmel. XILINX provides the TMRTool that automatically builds TMR into Xilinx space-grade FPGA designs, providing complete SEU and SET immunity, as well as, SEU controllers with a built-in readback CRC facility that detect whether an SEU has occurred in the configuration cells. Apart from this, for FPGA implementations of processor-based embedded space systems, self-test and diagnosis routines can be executed complementary to configuration memory scrubbing in order to ensure the normal operation during system start-up, periodically, or in case of error detection. Such routines are available for processor cores, co-processors, floating-point units, embedded memories, peripherals, etc.

At the system level are applied redundant-based fault-tolerant architectures depending on the cost and the available mass and power budget. The most common such architectures are: (a) duplex architecture (1 hot system and 1 cold system (spare), or 2 hot systems operating in parallel) for standard availability, (b) triplex architecture for high availability, (c) triplex architecture with spare(s), and (d) quadruplex architecture for Byzantine fault tolerance. Besides, error detecting and/or correcting codes are applied like parity codes for serial communication, cyclic redundancy check code for detecting burst errors in digital networks and storage devices, SEC-DED codes for memories, and Reed-Solomon codes as NASA standard. Dedicated low-density parity-check (LDPC) codes have been recently adopted by the committee for space data systems (CCSDS) recommended standard for telemetry and telecommand synchronization and channel coding. CCSDS has issued two classes of quasi-cyclic (QC) LDPC codes for telemetry applications: one for near-earth (C2) and another for deep-space communications (AR4JA), as well as, a set of short block length LDPC codes intended for telecommand applications. Apart from these, watchdog timers are used to recover the system from single event functional interrupts. All special circuits design techniques that can be applied at layout level, at architectural and design level, or at system level are referred as radiation hardening by design (RHBD).

In addition, latching current limiters (LCL) provide active overload protection of power lines in satellites. These devices are placed at the power input of any subsystem. They provide overload protections without generating dangerous voltage transients. In space applications sensitive to single event latch-up (SEL), they are mandatory in order to detect the leakage current increase and to rapidly recover it by switching off the power supply before devices get permanently damaged.

In case that hardware redundancy is limited or not affordable at all, software/time redundancy can be a viable solution to deal with nondestructive SEEs. The general idea is to execute the parts of the application software several times on the same processing unit before comparing the results. The key points of this methodology are a limited hardware overhead, but a significant performance overhead. Software redundancy can be achieved at instruction level, task level, and application level (where a hypervisor implements two virtual machines, each of them executes the program in its own address space, acquiring its set of data, processing it, and producing its set of results).

Finally, exposure to radiation environment of semiconductor devices can be reduced by shielding the circuit's package and/or the entire system. The vast majority of solar energetic particles are stopped by modest depths of shielding. However, Galactic Cosmic Rays (GCR), composed of highly charged and highly energetic particles, are much more challenging. Hydrogenous materials, such as polyethylene, have been shown to be more effective shields against GCR-like irradiation than aluminum.

2.3 Future Computing Systems for Space Applications

For years, ASICs, which can be made sufficiently radiation tolerant for space applications by using RHBD, were the only solution available to system designers for high-performance space applications. However, as the name ASIC implies, this approach has the drawback of long development time, high fabrication, and non-recurring engineering (NRE) costs, as well as, low adaptability. Today's SRAM-based FGPAs technology offers qualification for space applications, dynamic partial reconfiguration for in-flight adaptability, high density, and powerful hardwired blocks for enhanced DSP performance. Such FPGA technology offers unique advantages over both one-time programmable (OTP) FPGAs and ASICs and can be considered as an excellent platform for implementation of on-board PDPU functions, including high-performance image and data processing. Of course, due to the inherent SEE susceptibility of SRAM FPGAs, SEE mitigation techniques, such as TMR and configuration memory scrubbing must be employed for reliable operation.

A reconfigurable PDPU based on SRAM-based reconfigurable FPGAs offers many system-level advantages due to its ability to support planned, mode-dependent functional alterations, as well as, unplanned updates [26]. Operationally, the major system advantage is that it allows upgrades after launch, greatly enhancing mission profile and extending valuable system life time. These upgrades may be due to changing operational requirements, improved image and data processing algorithms, or in response to in-flight calibration or SEE mitigation. Reconfigurable PDPUs in space applications based on radiation-tolerant reconfigurable SRAM-based FPGAs have been already successfully demonstrated in the Venus Express Monitoring Camera (VMC) and the Framing Camera on NASA's DAWN mission and are considered for next ESA missions.

A dynamically reconfigurable payload data processing unit (DR-PDPU) can exploit, for space applications, the partial and dynamic reconfigurability of today's state-of-the-art SRAM-based FPGA technology. It is important to distinguish between rad-tolerant and radiation-hardened SRAM-based FPGAs. Rad-tolerant

FPGAs (such as Virtex 4QV) are rather sensitive to SEUs, while the radiation-hardened FPGAs (such as Virtex 5QV) are a lot more robust. In-flight dynamic adaptability enables multiple independent modes to be time multiplexed on the same processing resource (a space-qualified FPGA), allowing the hardware to be sized for the maximum operational load, rather than for the aggregate of every function, with attendant savings in mass, power, and design complexity. Furthermore, it allows run-time tailoring of the data processing algorithms (e.g., by switching between lossy and lossless compression algorithms), where the other parts of the dynamically reconfigurable FPGA remain operative. Therefore, in-flight dynamic adaptability using DR-PDPUs is a cutting-edge space technology, beyond the state of the art, and it is going to be demonstrated in space in the near future (e.g., polarimetric and helioseismic imager (PHI) instrument for ESA Solar orbiter mission [27]). Currently, two technology demonstrators are developed through the ESA's basic technology research programme (TRP). The first demonstrator [28] consists of at least two FPGAs: one dynamically, partially reconfigurable rad-tolerant FPGA (e.g., XILINX Virtex-4QV) for implementing the dedicated hardware on-board processing and one separate TMR by a flash reprogrammable FPGA (e.g., a Microsemi/Actel ProASIC3 FPGA) for implementing the system controller and the required space standard interfaces (see Fig. 5). In the actual mission, an Antifuse-based Microsemi RTAX device will replace ProASIC3. The second demonstrator [29] consists of the SpaceWire RTC AT7913E, one or more partially reconfigurable FPGAs (PR FPGAs), and a communication FPGA (COM FPGA), which is used to interconnect the SpaceWire RTC and the PR FPGAs. Both demonstrators have a configuration controller which is responsible for dynamic partial reconfiguration and configuration memory scrubbing of the reconfigurable FPGA.

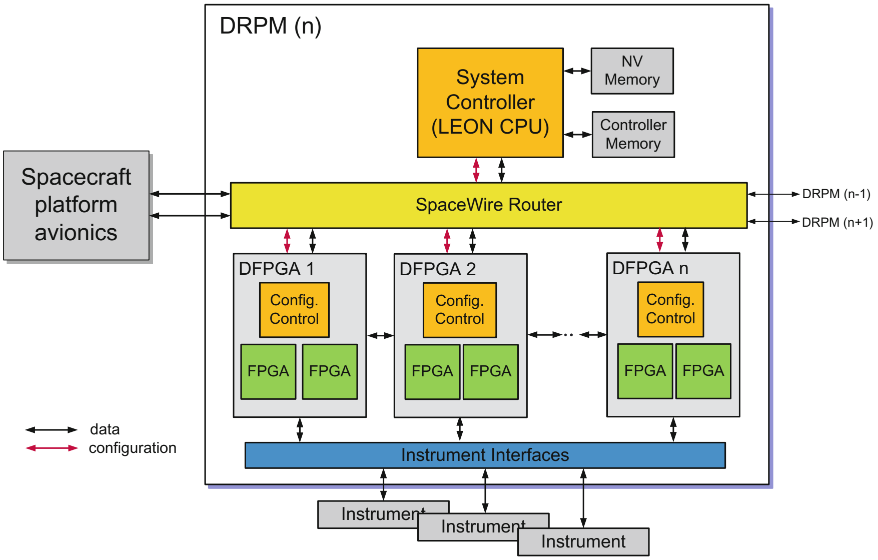


Fig. 5 First Demonstrator [28]

Although the state-of-the-art space-grade FPGAs have been proved well suitable to PDPU implementation, their use is restricted by Export Administration Regulations (EAR) from the U.S. Department of State. Hence, in order to address the identified needs, ESA is conducting additional R&D activities through three parallel main development routes which are not restricted by EAR [23]:

1. Hardening of COTS processors against radiation effects by dedicated hardware and software. This route offers comparatively short development time and moderate development cost. The recurrent cost is relatively high due to the extra (partially rad-hard) hardware and software needed for error mitigation. The achievable performances in areas such as low mass and power, reliability, and total dose tolerance are lower than those achievable with dedicated space-qualified chips that may be developed in routes 2 and 3.
2. Hardening of a proven COTS processor architecture by using a space-qualified ASIC platform and transparent design modifications. This route requires a high initial investment in IP licensing, chip level hardening and manufacturing/qualification process. Once the chip becomes available, it allows the development of reliable high-performance low mass data processing systems at low recurrent cost.
3. Development of a multi-core DSP/massively parallel IP-based processor. It includes a combination of powerful processor cores, network on chip (NoC) technologies, and on-chip memory elements can lead to processor architectures that are scalable, dependable, and high performance. This approach is expected to require a relatively high initial investment similar to (but lower than) option 2. Recurrent cost of systems based on a massively parallel processor chip is expected to be low.

Based on these development routes, the current development activities of ESA in nanoscale technology with respect to high performance, dependable multi-core next-generation payload processors are summarized as follows:

1. High-Performance COTS-Based Computer that follows development route 1 and is performed by Astrium France. The high-performance development will be based on COTS DSP chips, with radiation mitigation techniques implemented in a combination of hardware and software. Due to the modular design, the processing modules in use may be based on COTS DSP, general purpose processor (GPP), or FPGA technology according to user needs.
2. High Processing Power Digital Signal Processor That follows development route 1 and is performed by Astrium UK. Low mass and very low power consumption are among the driving design requirements, as they are derived from studies of future missions (Euclid, Plato, etc.).
3. European Digital Signal Processor Tradeoff and Definition Study that follows development route 2 and is performed by Astrium UK. It aims at the identification of the most promising DSP IP (among Analog Devices ADSP-21469, ATMEL Diopsis 940HF and Texas Instruments TMS320C6727B) for a possible migration step to a space-qualified ASIC platform. The migration of radiation

effects includes removal of IP parts that are not required for space, application of TMR, and EDAC techniques and addition of space standard interfaces like SpaceWire. A key requirement is to maximize the transparency of modifications to the software development environment (SDE) in order to avoid incompatibilities between commercial chips (which would be used in early hardware and software development phases) and the space-qualified version.

4. Massively Parallel Processor Breadboarding (MPPB) that follows development route 3 and is performed by Recore Systems BV NL. It aims at the development of a NoC-based system that combines two DSP cores (Xentium) with a LEON2 controller. It includes a set of space standard interfaces and features such as SpW including RMAP protocol support, CCSDS timers, ADC/DAC interfaces, and both on-chip and off-chip memories. The system has been developed on a XILINX Virtex-5 FPGA platform.
5. High-Performance Data Processor (HPDP) prototyping and performance assessment that follows development route 3 and is performed by Astrium Germany and ISD Greece. The HPDP core consists of a dynamically reconfigurable array processor on a radiation-tolerant ASIC technology (like ATC18RHA from ATMEL) providing high throughput data I/O paths, resulting in a processor having high performance and low power consumption. The potential of the HPDP lies in the processing capability of high data volumes in the signal processing domain, especially where flexibility and in-orbit programmability or reconfiguration is required. The development of HPDP has high strategic and technological importance for Astrium as it deals with the development of on-board programmable payloads for telecommunication and earth-observation applications characterized by their ability to sustain a high data throughput combined with a high performance level and the flexibility to adapt to emerging standards and improvements.

To conclude, we emphasize that in the commercial world, multi-core processors are becoming dominant, and it is expected that space will follow this trend the next years. As an example we refer the ESA next-generation microprocessor (NGMP) which is a fault tolerant multiprocessor system-on-a-chip (SoC), based on four LEON4 cores. Downsized configurations of the design are currently provided by Aeroflex Gaisler, as FPGA prototypes and development boards. Further development phases will cover the manufacturing and validation of prototypes in space ASIC technology and manufacturing and qualification of flight parts. The microprocessor chip development is also complemented by activities in the SW field.

3 Fault Tolerance for Medical Devices

In this section, we consider life-saving devices and highly reliable/low-power systems supporting wearable sensors. This is a very challenging domain from dependability point of view, yet very different from transportation and space.

Life-saving devices such as, for example, implanted pacemakers and defibrillators, heartbeat monitors, and glucose-level monitors require high levels of reliability. In this context, the workgroup activities will focus on the assessment of fault mechanisms affecting life-saving devices, test methods, and on related fault tolerant design techniques.

Power consumption is a very critical factor for portable medical devices, including wearable health sensors. In particular the need for energy saving in digital systems is pervasive, and the interaction with reliable design will increase in the next years. The workgroup activities will focus on the interaction between reliability and low power requirements and will try to extend the achieved results to more general application scenarios.

3.1 Manufacturability- and Reliability-Related Challenges

Medical devices have ultrahigh reliability requirements as a device failing in the field can be life threatening. From an integrated device perspective a wide variety of functional, structural and current-based tests are deployed in combination with lifetime and burn-in tests.

Products themselves are subject to a variety of regulations and requirements as illustrated below. Figure 6 lists as an example of requirements and regulations affecting a pacemaker.

3.2 Current Practices

Current practices in the design, development, and realization involving a combination of design for test (DFT), design for reliability (DFR), fault tolerant device

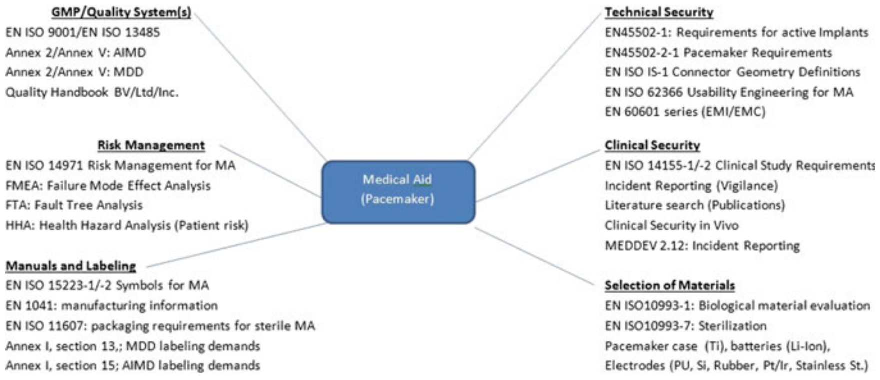

GMP/Quality System(s)
EN ISO 9001/EN ISO 13485
Annex 2/Annex V: AIMD
Annex 2/Annex V: MDD
Quality Handbook BV/Ltd/Inc.
Risk Management
EN ISO 14971 Risk Management for MA
FMEA: Failure Mode Effect Analysis
FTA: Fault Tree Analysis
HHA: Health Hazard Analysis (Patient risk)
Manuals and Labeling
EN ISO 15223-1/-2 Symbols for MA
EN 1041: manufacturing information
EN ISO 11607: packaging requirements for sterile MA
Annex I, section 13,; MDD labeling demands
Annex I, section 15; AIMD labeling demands
Medical Aid
(Pacemaker)
Technical Security
EN45502-1: Requirements for active Implants
EN45502-2-1 Pacemaker Requirements
EN ISO IS-1 Connector Geometry Definitions
EN ISO 62366 Usability Engineering for MA
EN 60601 series (EMI/EMC)
Clinical Security
EN ISO 14155-1/-2 Clinical Study Requirements
Incident Reporting (Vigilance)
Literature search (Publications)
Clinical Security in Vivo
MEDDEV 2.12: Incident Reporting
Selection of Materials
EN ISO10993-1: Biological material evaluation
EN ISO10993-7: Sterilization
Pacemaker case (Ti), batteries (Li-Ion),
Electrodes (PU, Si, Rubber, Pt/Ir, Stainless St.)

Fig. 6 Medical Aid (Pacemaker) product requirements

techniques in combination with stringent IC test schemes involving a wide variety of functional, structural, and current-based tests are deployed in combination with lifetime and burn-in tests. From a current-test perspective this implies the use of advanced IDDQ and ISSQ test schemes, dynamic and transient (IDDT) test methods and power consumption verification and validation involving power-down observations and techniques such as the energy consumption ratio (ECR) test method. Failing devices are subject to extensive failure analysis to determine the root cause for failure and screen for process or design weaknesses.

Once deployed in the field, the DFR and fault tolerant design additions are used to track the health of a device and send out warning signals. As an example of a mitigation approach for implantable devices: If during field operation a faulty behavior of the system is recognized, there is little the device can do but to reset and go into an assumed safe state and send a signal to the doctor. The doctor calls the patient in and examines. In most cases so far, when the device alerts the doctor, it has been the result of the patient having had an episode which the device feels the doctor should know about or there is a low battery alert. In the first case, it is the patient needing attention. In the second, nothing can be done but have the patient subject to surgery either to replace batteries or in the worst case to replace the implant.

When developing implantable devices, much attention is paid to properties such as fault tolerance, protection against external influences such as for example electromagnetic interferences and achieving a long lifetime, preferably with minimal or no maintenance actions. Software aspects came into the picture triggered by the Therac-25 incident: "*A radiation therapy device malfunctions and delivers lethal radiation doses at several medical facilities. Based upon a previous design, the Therac-25 was an "improved" therapy system that could deliver two different kinds of radiation: either a low-power electron beam (beta particles) or X-rays. The Therac-25's X-rays were generated by smashing high-power electrons into a metal target positioned between the electron gun and the patient. A second "improvement" was the replacement of the older Therac-20's electromechanical safety interlocks with software control, a decision made because software was perceived to be more reliable.*

What engineers didn't know was that both the 20 and the 25 were built upon an operating system that had been kludged together by a programmer with no formal training. Because of a subtle bug called a "race condition", a quick-fingered typist could accidentally configure the Therac-25 so the electron beam would fire in high-power mode but with the metal X-ray target out of position. At least five patients die; others are seriously injured."

The general safety practice for the development of medical systems is presented in the IEC 62304 safety standard for medical software. A number of European regulations cover both medical equipment and implantable devices (starting with the basic directive 93/43/EEC and ending with the more recent directives for implantable devices 2007/47/EC and 2013/C 22/01).

Fault tolerance in medical applications can be traced back to the early 80s, where the focus was on designing of completely fault tolerant software and hardware for medical systems. For example, McAllister and Nagle [30] have published an article "*Toward a fault-tolerant processor for medical applications*" arguing for medical systems with embedded fault tolerance support in both software and hardware. An interesting and still valid summary of faults in medical devices can be found in the survey made by Wallace and Kuhn [31]: "*Lessons from 342 medical device failures*." In the recent years, the research focus has shifted to verifiable fault tolerance techniques and trade-offs between fault tolerance and other constraints, for example, power. In the article "*The System-Level Simplex Architecture for Improved Real-Time Embedded System Safety*" Bak et al. [32] describe a study on a pacemaker application with fault tolerance. Also Maheshwari et al. [33] in their paper on "*Trading off transient fault tolerance and power consumption in deep submicron (DSM) VLSI circuits*" use a pacemaker as a safety-critical application in their study of trade-offs.

One of the most recent advances in fault tolerance of medical devices cover microfluidic biochips and the ways to overcome different types of faults with mixers and leakages is described the paper by Chakrabarty and Su [34]: "*Design of fault-tolerant and dynamically-reconfigurable microfluidic biochips.*" With the increased inter-connectivity of medical devices, a detection and identification of faults in the distributed medical environment becomes a critical issue as addressed by Xiong et al. [35] in "*Comparative analysis of quality of service and memory usage for adaptive failure detectors in healthcare systems.*" Fault tolerant processors and hardware issues in medical devices have recently gained attention again. For example, Seepers et al. [36] in "*Architecture-level fault-tolerance for biomedical implants*" discuss a novel RISC architecture for fault tolerance in hardware of medical devices, taking into account power consumption and silicon area.

Patel et al. [37], in their review paper "*A review of wearable sensors and systems with application in rehabilitation*" summarize recent developments in the field of wearable sensors and systems that are relevant to the field of rehabilitation. The growing body of work focuses on the application of wearable technology to monitor older adults and subjects with chronic conditions in the home and community settings justifies the emphasis of this review paper on summarizing clinical applications of wearable technology currently undergoing assessment rather than describing the development of new wearable sensors and systems. A short description of key enabling technologies (i.e., sensor technology, communication technology, and data analysis techniques) that have allowed researchers to implement wearable systems is followed by a detailed description of major areas of application of wearable technology. Applications described in this review paper include those that focus on health and wellness, safety, home rehabilitation, assessment of treatment efficacy, and early detection of disorders. The integration of wearable and ambient sensors is discussed in the context of achieving home monitoring of older adults and subjects with chronic conditions. Future work required to advance the field toward clinical deployment of wearable sensors and systems is discussed.

3.3 Future Perspectives

In the coming nanoscale era, chips are becoming less reliable, while manufacturing reliable chips is becoming increasingly more difficult and costly. Prominent causes for this are the shrinking device features, the growing rapidly number of components on a given area of silicon, as well as the increasing complexity of current and future chips. It is expected that a significant number of devices will be defective already at manufacture time and many more will degrade and fail within their expected life time. Furthermore, process variations as well as the increasing number of soft errors introduce additional sources of errors for future chips.

The ITRS targets a constant defect rate (1395 defects/m^2) in order to keep the chip yield constant. Such a target is expected to substantially increase the chip manufacturing cost of future semiconductor technologies. Alternatively, chips need to be designed to tolerate an increasing number of defects in order to maintain a high yield. Apart from defects at manufacture time, aging effects are becoming more severe leading to more permanent and intermittent faults during the lifetime of a chip. Transistors degrade faster; while the degradation rate is further accelerated by the heavy testing processes (e.g. burn-in). Aging is expected to shorten SoC lifetime and to be a significant source of errors in technologies beyond 16-nm. Process variations cause devices to operate differently than expected; such variations are random dopant fluctuations, heat flux, as well as lithography problems due to the shrinking geometries. Currently, on-chip clock frequency and total power consumption present variations up to 30 and 50%, respectively, across different parts of a single chip; it is projected that variations will only become more severe in the future and worst case, deterministic design will be insufficient and unable to deliver reliable systems. Finally, as transistor count increases, the number of soft errors on a chip (i.e. transient faults) grows exponentially. For example, by the 16-nm generation, the failure rate will be almost 100-fold higher that at 180-nm; current fault tolerance techniques such as simple check-pointing will, then, incur prohibitively high energy and performance costs.

DeSyRe

As feature size continues to shrink and chips become less reliable, the cost for delivering reliable chips is expected to grow for future technology nodes. The price in system power and energy consumption, in performance degradation, and in extra resources, is getting higher in order to perform redundant computations in time or in space. However, it is a well-known fact that power consumption is becoming a severe problem, while performance no longer scales very well (mostly due to power-density limitations). To address some of these aspects, as an example, the EU-funded *DeSyRe project* (FP7 STREP—www.desyre.eu) aimed at reliable systems containing and tolerating unreliable components rather than targeting totally fault-free systems. The project's goal was to describe a new, more efficient design framework for SoCs which provides reliability at lower power and performance cost.

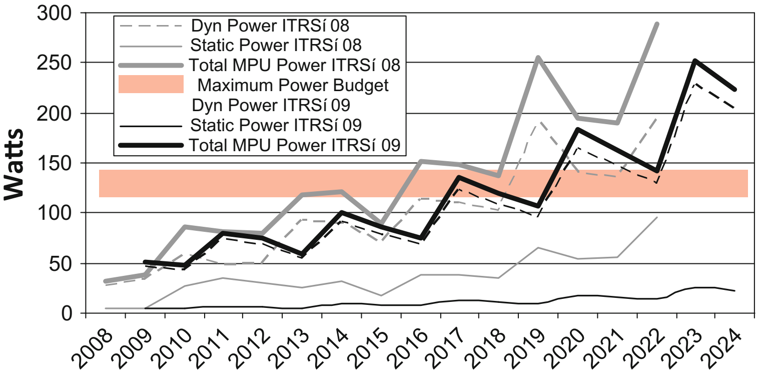


Fig. 7 Dynamic, static and total power consumption of a Microprocessor chip versus the maximum available power budget, based on the ITRS 2008 and 2009 projections (*Courtesy* DeSyRe project)

Although the above technology trends make the design of future SoCs harder, one of them can be turned to our advantage. As shown in Fig. 7, the increasing power-density limits the gate density. In a few years, significant parts of a chip will be forced to remain powered-down in order to keep within the available power budget. The DeSyRe project proposed to exploit the aforementioned unused resources to offer flexibility and configurability on a chip. Until now, reconfigurable hardware had a significant resource overhead; this limitation no longer exists as on-chip resources are becoming "cheaper". A dynamically reconfigurable hardware-substrate can provide an excellent solution for defect tolerance; it can be used to adapt to faults on demand, isolate and correct defects, as well as to provide spare resources to substitute defective blocks.

In the DeSyRe project, the intention was to use such a reconfigurable substrate and combine it with system-level techniques to provide adaptive and on-demand reliable systems. The cutting- edge medical systems have been prototyped on the envisioned DeSyRe SoC architecture: (i) a high-performance, portable system for real-time olivocerebellar (i.e., brain) simulations; and (ii) a low-power, wearable (and, long-term, implantable) artificial-pancreas system for automatic blood-glucose regulation in diabetics.

Next-generation implantable neurostimulators devices

Medical treatment of brain disorders and diseases with electric stimulation has progressed from crude electroshocks given to patients in the insanity institutes to using very low-voltage or low-amplitude electric pulses through well-positioned electrodes in the brain. Such treatments have an impressive effect on a broad range of diseases: Tinnitus and auditory hallucinations, Tourette's syndrome, obsessive compulsive syndrome, epilepsy, Alzheimer, Parkinson, and migraine, to name a few. The first applications appeared in the market and currently an impressive number of patients are "wearing" so-called neurostimulators in their brains.

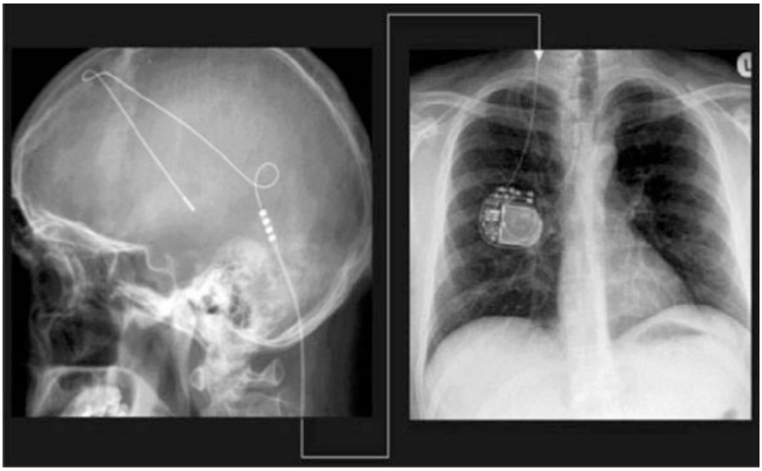


Fig. 8 Example of state of the art neurostimulator. Implant is located in the chest cavity (*right*); stimulating electrode is located deep in the brain (*left*) (*Courtesy* Braininnovations.nl)

However, despite their well-appreciated contribution to the health of the patients, these devices are still very primitive and uncomfortable, and not meeting the current state of technologies. For example, they need big batteries for coping with their energy requirements, which have to be implanted somewhere else in the body, with cables leading through the neck to the actual electrodes implanted in the brain (Fig. 8). The cables tend to break sooner or later, or adhere to the patient's tissue, causing pain and discomfort. Further, the current implant pulse generators can only vary wave amplitude, duration, and frequency, which is hardly better than the parameters of an electric blinking light. In reality, electricity has a lot more parameters to tune, such as wave form and wave pattern, which have been shown to influence the effects on the brain in a different way. Last but not least, the currently available neurostimulators offer no feedback options. The reactions of a patient's brain to an electric pulse can only be measured implicitly, i.e., through reactions of the patient or the alpha waves emitted by the brain. So, in practice, it is the doctor who adapts and optimizes the parameters, not the device, which is of little use to free-roaming patients.

With the above limitations in mind, the time for a new, intelligent neurostimulator, equipped with a small, low-power, wirelessly accessible controller and the relevant feedback sensors, which automatically responds to the patient's reactions is needed. The device must be fed by a compact long-life battery implanted in the brain without cables, rechargeable from outside the body, or (in the long run) even by the body itself. The neurostimulator control unit needs to be versatile in its stimulation patterns to deal with the particular patient idiosyncrasies and with the habituation of the brain to the neurostimulator. Such requirements, of course, introduce additional device complications; more electrodes with wider output-current swing, more intensive on-board processing, versatile functionality, more unpredictable battery life and, thus, high needs for dynamic power management, etc. are needed. Multiple electrodes will allow for clinical testing of various hypotheses such as “reconditioning neurostimulation”, which involves manipulating the neurocircuitry of addiction to, e.g. cure alcoholics. Preliminary prototypes of such a neurostimulator are currently being built and clinically assessed under the *SINS theme (Small Implantable Neurostimulators)* of the *BrainInnovations Consortium* (www.braininnovations.nl), a cross-disciplinary research consortium comprising engineers and medical doctors, among other specialties.

Human++

It is anticipated that nanoelectronics and microsystems technology will increase the functionality of lifestyle and healthcare devices to gradually match the needs of society. The technology will enable people to carry their personal body-area network (BAN) that provides medical, lifestyle, assisted living, sports, or entertainment functions for the user, without visible interference with their active lives. Prevention rather than detection and cure will be the future paradigm.

The successful realization of this vision requires innovative solutions to remove the critical technological obstacles of size and power. The overall size should be compatible with the required form factor. This requires new integration and packaging technologies. The energy autonomy of current battery-powered devices is limited and must be extended. Intelligence should be added locally so that each sensor is capable of storing, processing and transferring data continuously or on a need/event-triggered basis. The energy consumption of all building blocks needs to be drastically reduced to allow energy autonomy.

The technology parts of IMEC’s HUMAN++ program (http://www.imec.be/ScientificReport/SR2011/1414066.html) addressed the key technology building blocks required to enable this vision. The power challenge impacts on the complete system design of the node ranging from the sensor (addressing Ultralow-power sensors) and its front-end (addressing High resolution low bandwidth analog-to-digital converter (ADC)) to the signal processing (addressing Ultralow-power digital signal processing (ULP-DSP)), the wireless communication (addressing Ultralow-power radio) and the power generation and storage (addressing Micro power generation and storage). The second challenge relates to appropriate biological and physiological sensing where aside from achieving overall low power, new sensing mechanisms are needed as well as suitable algorithms for information extraction. The BAN technology-integration program aims at materializing these ingredients into prototypes and deploying these prototypes in real-life environments. The aim is achieving reliable, noninvasive and long-term monitoring of physiological and biological signals on-the-move.

Three application fields are enabled by the BAN research and they concern the health patch, the wearable electroencephalography (EEG) and emotion monitoring. The health patch aims at developing a wearable, multi-modal, and reliable patch for ambulatory health monitoring applications. With its industrial partnership, IMEC created an ecosystem with actors across the value chain, from chip manufacturers, material and skin adhesive companies, system integrators, telecommunication companies, and medical devices.

The wearable EEG aims at taking EEG monitoring technologies out of the lab or hospital environment, while maintaining a signal quality that is comparable to lab equipment. It addresses the challenges associated with using EEG in the home environment, and eventually in ambulatory conditions. IMEC's ecosystem involves clinical partners for the definition of the requirements, and the evaluation of the prototypes. The emotion monitor R&D concerns the use of BAN for monitoring emotions and stress, and targets the deployment of these technologies out-of-the lab, in daily-life situations. Innovation is needed at the system, algorithm and application level.

Acknowledgements Authors would like to thank Christos Strydis, Ioannis Sourdis, Jørgen Ilstad and Oliver Bringmann for their valuable contributions during preparation of this chapter.

References

1. C. Hernandez, J. Abella, Timely error detection for effective recovery in light-lockstep automotive systems, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **34**(11), 1718–1729 (2015)
2. MICROSAR Safe: Safety According to ISO 26262 up to ASIL D—Compatible with AUTOSAR. https://www.tttech.com/products/automotive/autosar-safety-software/microsar-safe/. Accessed 7 July 2016
3. V. Izosimov, Scheduling and optimization of fault-tolerant distributed embedded systems. Doctor Thesis No. 1290, Dept. of Computer and Information Science, Linköping University, Sweden (2009)
4. G. Lagunoff, BorgWarner eAWD. http://hybridfordonscentrum.se/wp-content/uploads/2014/05/20140404_BorgWarner.pdf. Accessed 7 July 2016
5. AUTOSAR 4.0. https://www.autosar.org/specifications/release-40/. Accessed 7 July 2016
6. ISO 26262:2011 Road vehicles—Functional safety (2011)
7. D. Reinhardt, G. Morgan, An embedded hypervisor for safety-relevant automotive E/E-systems, in *9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)* (2014)
8. What is Docker? https://www.docker.com/what-docker. Accessed 7 July 2016
9. V. Izosimov, G. Di Guglielmo, M. Lora, G. Pravadelli, F. Fummi, Z. Peng, M. Fujita, Time-constraint-aware optimization of assertions in embedded software. J. Electron. Test. Theory Appl. (JETTA) **28**(4), 469–486 (2012)
10. V. Izosimov, I. Polian, P. Pop, P. Eles, Z. Peng, Analysis and optimization of fault-tolerant embedded systems with hardened processors, in *Design Automation and Test in Europe (DATE 2009)* (Nice, France, 2009), pp. 682–687
11. E. Dubrova, *Fault-Tolerant Design* (Springer, 2013)
12. V. Izosimov, A. Asvestopoulos, O. Blomkvist, M. Törngren, Security-aware development of cyber-physical systems illustrated with automotive case study, in *Design, Automation and Test in Europe (DATE 2016)* (2016), pp. 818–821
13. AEC-Q100/AEC-Q200—Automotive Electronics Council. http://www.aecouncil.com/AECDocuments.html. Accessed 7 July 2016

14. ZVEI, Handbook for Robustness Validation of Semiconductor Devices in Automotive Applications. http://www.zvei.org/Publikationen/Robustness-Validation-Semiconductor-2015.pdf. Accessed 7 July 2016
15. G. Jerke, A.B. Kahng, Mission profile aware IC design—A case study, in *Design Automation and Test in Europe (DATE 2014)* (Dresden, 2014), pp. 1–6
16. EN 50126-1 Railway applications—The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)—Part 1: Basic requirements and generic process
17. EN 50128 Railway applications—Communications, signalling and processing systems—Software for railway control and protection systems
18. EN 50129 Railway applications—Communication, signalling and processing systems—Application Guide for EN 50129—Part 1: Cross-acceptance
19. R.D. Schrimpf, D.M. Fleetwood (eds.), *Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices* (World Scientific, Singapore, 2004)
20. ECSS-E-HB-10-12A, Space engineering, Calculation of radiation and its effects and margin policy handbook
21. ECSS-Q-ST-30C, Space product assurance—Dependability
22. ECSS-Q-ST-40C, Space product assurance—Safety
23. R. Trautner, ESA'S roadmap for next generation payload data processors, in *Proceedings of the 2011 Data Systems In Aerospace (DASIA)* (Malta, 2011)
24. ESA Round Table Synthesis, Next Generation Processor for On-board Payload Data Processing Application (2007)
25. ECSS-Q-HB-60-02A DIR2.6, DRAFT, 30, Space engineering, product assurance—Techniques for Radiation Effects Mitigation in ASICs and FPGAs, October (2015)
26. P.J. Pingree, Advancing NASA's on-board processing capabilities with reconfigurable FPGA technologies: opp & implications, in *IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, April (2010)
27. B. Fiiethe et al., Adaptive hardware by dynamic reconfiguration for the Solar Orbiter PHI instrument, in *Proceedings of the NASA/ESA Conference on Adaptive Hardware Systems*, June (2012)
28. F. Bubenhagen et al., Enhanced dynamic reconfigurable processing module for future space applications, in *Proceedings of the International Space Wire Conference (ISC)* (2010)
29. F. Dittmann et al., Implementation of a dynamically reconfigurable processing module for SpaceWire Networks, in *Proceedings of the International Space Wire Conference (ISC)* (2010)
30. D.F. McAllister, D.F. Nagle, Toward a fault-tolerant processor for medical applications, in *Proceedings of the Symposium on the Engineering of Computer-Based Medical Systems* (1988), pp. 101–104
31. D.R. Wallace, D.R. Kuhn, Lessons from 342 medical device failures, in *4th IEEE International Symposium on High-Assurance Systems Engineering* (1999), pp. 123–131
32. S. Bak et al., The system-level simplex architecture for improved real-time embedded system safety, in *Real-Time and Embedded Technology and Applications Symposium* (2009), pp. 99–107
33. A. Maheshwari, W. Burleson, R. Tessier, Trading off transient fault tolerance and power consumption in deep submicron (DSM) VLSI circuits. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **12**(3) (2004)
34. K. Chakrabarty, F. Su, Design of fault-tolerant and dynamically-reconfigurable microfluidic biochips, in *DATe 2005*, vol. 2 (2005), pp. 1202–1207
35. N. Xiong et al., Comparative analysis of quality of service and memory usage for adaptive failure detectors in healthcare systems. IEEE J. Sel. Areas Commun. **27**(4), 495–509 (2009)
36. R.M. Seepers, C. Strydis, G.N. Gaydadjiev, Architecture-level fault-tolerance for biomedical implants, in *International Conference on Embedded Computer Systems (SAMOS)* (2012), pp. 1202–1207
37. S. Patel, H. Park, P. Bonato, L. Chan, M. Rodgers, A review of wearable sensors and systems with application in rehabilitation. J. NeuroEng. Rehabil. **9**, 21 (2012)

Part III
State of the Art and Vision

Variation-Mitigation for Reliable, Dependable and Energy-Efficient Future System Design

Shidhartha Das

Abstract Integrated circuits in modern SoCs and microprocessors are typically operated with sufficient timing margins to mitigate the impact of rising process, voltage and temperature (PVT) variations at advanced process nodes. The widening margins required for ensuring robust computation inevitably leads to conservative designs with unacceptable energy-efficiency overheads. Reconciling the conflicting objectives imposed by variation-mitigation and energy-efficient computing will require fundamental departures from conventional circuit and system-design practices. We begin by reviewing how energy-efficiency constrains computing across the entire spectrum, from ultra-low-power sensor node systems to high-performance supercomputing systems delivering peta-flop order performance. We discuss how rising variations adversely impact energy-efficient system design in the traditional method of designing for the worst case. We classify various sources of variation and discuss the traditional approaches for variation-mitigation and their limitations. The latter half of the chapter deals with several promising techniques for variation-mitigation. We discuss in situ ageing monitors, error-resilient techniques and adaptive-clocking techniques that aim at improving system-efficiency by actively reducing design guardbands. In particular, we focus on error-resilient techniques that exploit tolerance to timing errors to automatically compensate for variations and dynamically tune a system to its most efficient operating point. We present the Razor approach as a pioneering example of such a technique. We present silicon measurement results from multiple industrial and academic demonstration systems that employ Razor dynamic voltage and frequency management. Finally, we conclude the chapter with few pointers on alternative techniques for variability-mitigation.

S. Das (✉)
ARM Research, Cambridge, UK
e-mail: shidhartha.das@arm.com

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_7

1 Introduction

It is a well-known observation that traditional feature-size scaling is increasingly running into fundamental physical limits. Technology innovations such as FinFETs and 3D stacking continue to deliver increased transistor densities. However, rising PVT variations combined with limited supply-voltage scaling significantly undermine automatic energy-efficiency gains traditionally obtained through process scaling. This has created a design paradox often referred to as “Dark Silicon” [1]: more gates can now fit on a die, but cannot actually be used due to strict power limits. Indeed, energy-efficiency is a first-class design constraint across the entire spectrum of computing.

Figure 1 presents a conceptual representation of the computing spectrum and highlights the importance of energy-efficiency in current-generation computing systems. Efficiency constraints are intuitively simpler to visualize for ultra-low-power computing systems at the lowest end of the power-performance spectrum. Such systems typically find usage in applications such as continuous health and infrastructure monitoring where form-factor constraints restrict the capacity of the battery used to power such systems. Operating under heavily energy-constrained environments often requires resorting to energy-harvesting techniques to make up for the energy shortfall. Mobile computers such as smartphones and laptop computers are also energy-constrained systems since they operate under a fixed power budget and typically have to provide several hours of compute-time for every charge cycle. Such systems also operate under strict thermal constraints, which limit the maximum power dissipation of such systems. At the very high-end of the spectrum are enterprise server systems and supercomputing systems that are wall-powered. Such systems are essentially power-constrained system since the

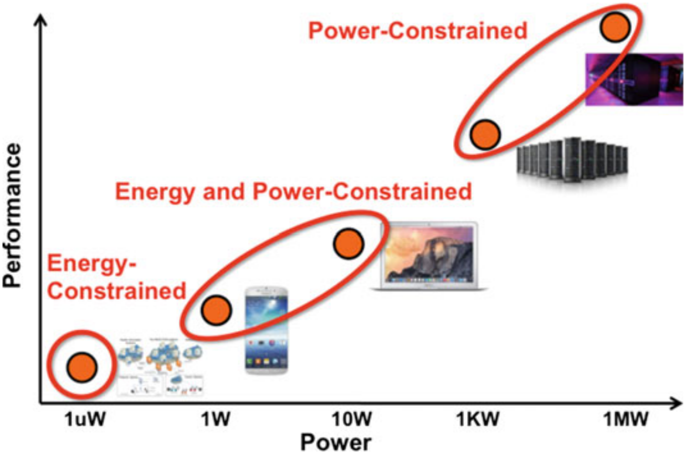


Fig. 1 Conceptual representation of the power-performance spectrum of computing—highlights the importance of energy-efficiency as a first-class design constraint

peak power consumption on these devices limits their Total Cost of Ownership (TCO). Thus, computing is essentially either power- or energy-constrained across the entire spectrum.

Moore's law-based dimensional scaling traditionally provided the necessary efficiency demand. However, sustained process scaling is now no longer economically feasible due to fundamental physical barriers. A direct consequence of smaller geometries and higher integration levels is that the manufacturing process is poorly controlled, leading to large variations in transistor performance. Susceptibility to single-event upsets and ageing-induced reliability issues that are increasingly pronounced at smaller geometries further exacerbate transistor variability. This makes systems susceptible to timing-failures due to gradual slow-down in transistor switching speeds, eventually leading to permanent functional failure.

The traditional approach of robust and reliable computing in the presence of variation relies upon operation at higher supply voltage and/or lower operating frequency. Addition of generous guardbands incurs significant power and performance overheads. Furthermore, operation at higher supply voltages leads to accelerated ageing, thereby impacting long-term system reliability.

Recently, error-resilient techniques have been proposed that mitigate the power, performance and reliability impact of excessive design margining. In lieu of margins, such techniques rely upon error-detection and correction techniques to reduce or eliminate voltage guardbands, leading to energy-efficient operation. Razor [2–5] is a specific example of such a technique where error-detecting circuitry at critical-path endpoints flag timing violations. Error-correction is achieved either through correct data substitution or through instruction-replay from a check-pointed state. In situ error-detection circuits and microarchitectural recovery mechanisms eliminate these margins and scale the supply voltage to the Point of First Failure (PoFF) and below. Error-detection and recovery enables Razor systems to survive both fast-moving and transient events, and adapt to the slow-changing prevailing conditions, allowing excess margins to be reclaimed. The reclaimed margins can be traded-off for per-device improvements in energy-efficiency or parametric yield improvement for a batch of devices.

Razor-enabled dynamic adaptation has demonstrated substantial improvements in performance and energy-efficiency in microprocessor pipelines. In [4], we demonstrate 52% energy savings at 1 GHz operation for a Razor-based ARM ISA processor. Related work in [5] shows 32% throughput improvements at iso-voltage and 17% voltage reduction at iso-frequency operation.

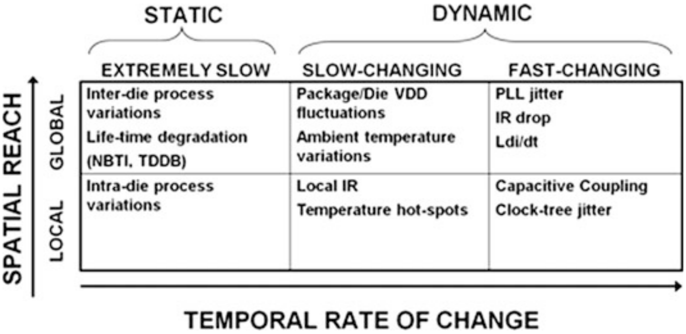
In this chapter, we review how variation-mitigation can be an effective tool for energy-efficient computing. In the following section, we classify the various sources of on-chip variation into their time rate of change and according to their spatial reach. In Sect. 3, we examine tracking circuits as a technique for compensating slow-changing variations. Section 4 examines various flavours of error-resilient computing. Section 5 discusses adaptive-clocking techniques. Finally, we end the chapter in Sect. 6 where we provide concluding remarks.

2 Classification of Variations

Figure 2 classifies the various sources of variations according to their spatial reach and temporal rate-of-change. Based on their spatial reach, variations can be *global* or *local* in extent. Global variations affect all transistors on die such as inter-die process variations and ambient temperature fluctuations. In contrast, local variations affect transistors that are in the immediate vicinity of one another. Examples of local variations are intra-die process variations, local resistive (IR) drops in the power-grid and localized temperature hot spots.

Based on their rate-of-change with time, variations can be classified as being *static* or *dynamic*. Static variations are essentially fixed after fabrication such as process variations, or manifest extremely slowly over processor lifetime such as ageing effects. Dynamic variations affect processor performance at runtime. Slow-changing variations such as temperature hot spots and board-parasitic induced regulator ripple have kilohertz time constants. Fast-changing variations such as inductive undershoots in the supply voltage can develop over a few processor cycles. The rate and the duration of these Ldi/dt droops is a function of package inductance and the on-chip decoupling capacitance. Coupling noise and Phase-Locked Loop (PLL) jitter are examples of local and extremely fast dynamic variations with duration less than a clock-cycle.

In general, slow-changing and global effects such as process variations and ageing effects are relatively easier to predict and compensate for. For instance, binning is a well-known technique to compensate for inter-die process variations. In contrast, fast-changing and local-effects do not provide sufficient temporal and spatial context necessary for compensation. As such, guardbanding is the only way to compensate for such effects. Example for such high-frequency and localized events are capacitive coupling effects and clock tree jitter.



	STATIC	DYNAMIC	
	EXTREMELY SLOW	SLOW-CHANGING	FAST-CHANGING
GLOBAL	Inter-die process variations Life-time degradation (NBTI, TDDB)	Package/Die VDD fluctuations Ambient temperature variations	PLL jitter IR drop Ldi/dt
LOCAL	Intra-die process variations	Local IR Temperature hot-spots	Capacitive Coupling Clock-tree jitter

Fig. 2 Sources of variations—taxonomy

In the subsequent sections, we provide an analysis of several variation-mitigation techniques including tracking circuits, error-resilient computation and adaptive-clocking techniques.

3 Tracking Circuits for Variation-Mitigation

Traditional adaptive techniques [6–13] based on canary or tracking circuits can compensate for certain manifestations of PVT variations that are global and slow changing. These circuits are used to tune the processor voltage and frequency taking advantage of available slack. Tuning is limited to the point where delay measurements through the tracking circuits predict imminent processor failure.

These circuits are limited by measurement uncertainty, the degree to which current and future events correlate and the latency of adaptation. Substantial margining for fast-moving or localized events, such as Ldi/dt, local IR-drop, capacitive coupling, or PLL jitter must also be present to prevent potential critical-path failures. These types of events are often transient, and while the pathological case of all occurring simultaneously is extremely unlikely in a real system, it is impossible to rule this out. Tracking circuits also incur significant calibration overhead on the tester to ensure critical-path coverage over a wide range of voltage and temperature conditions. The delay impact of local variations and fast-moving transients worsens at advanced process nodes due to aggressive minimum feature lengths and high levels of integration. This undermines the efficacy of tracking circuits.

Synthesized and automatically placed-and-routed designs present even greater challenges. Figure 3 highlights critical-paths on a Cortex-A9 core converging on a single critical-path endpoint. There are in excess of 100 paths within 70 ps of the critical-path at 90 nm technology. These paths cover 377 unique instances and 118 unique cell-masters, thereby making the problem of creating tracking paths extremely difficult.

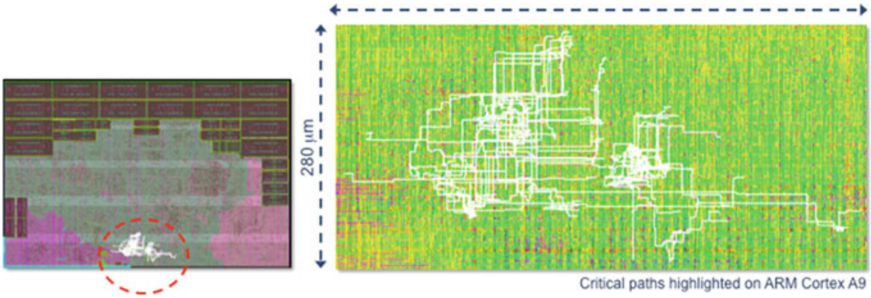


Fig. 3 Critical-paths on the ARM Cortex-A9 illustrating the complexity of creating suitable tracking circuits for synthesized and placed-and-routed designs

4 Error-Resilient for Variation-Mitigation—Razor

In contrast with the traditional adaptive techniques, the Razor approach [2–5] exploits the observation that the pathological combination of worst-case variation conditions occur extremely rarely in practice. Therefore, in Razor, requisite margins are added to the operating point dynamically according to the workload, prevailing environmental and silicon conditions.

In Razor, we exploit the dynamic nature of variations to speculatively operate a processor without statically added timing guardbands. Speculative operation requires efficient circuitry for reliable detection of and subsequent recovery from timing violations. A combination of error-detecting circuits and microarchitectural recovery mechanisms create a system that is robust in the face of timing errors, and can be tuned to an efficient operating point by dynamically eliminating unused guardbands.

The operational principle of Razor is illustrated in Fig. 4 and shows the qualitative relationship between the supply voltage, energy consumption and pipeline throughput of a Razor-enabled processor [3]. The PoFF of the processor (V_{ff}) and the minimum allowable voltage of traditional DVS techniques (V_{margin}) are also labelled in the figure. V_{margin} is much higher than V_{ff} under typical conditions, since safety margins need to be included to accommodate for worst-case operating conditions. Razor relies on in situ error-detection and correction capability to operate at V_{ff}, rather than at V_{margin}. The total energy of the processor (E_{tot}) is the sum of the energy required to perform standard processor operations (E_{proc}) and the energy consumed in recovery from timing errors ($E_{recovery}$). Of course, implementing Razor incurs power overhead such that the nominal processor energy (E_{nom}) *without* Razor technology is slightly less than E_{proc}. This overhead is

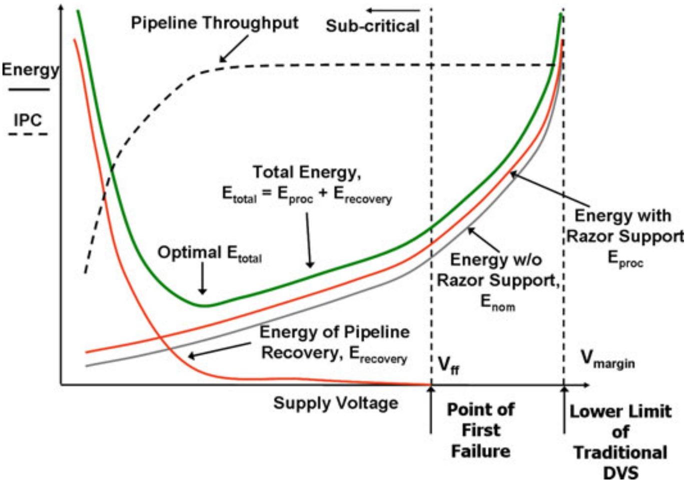


Fig. 4 Razor operational principle

attributed to the use of delay-error tolerant flip-flops on the critical-paths and the additional recovery logic required for Razor. However, since the extra circuitry is deployed only for those flip-flops that have critical-paths terminating in them, the power overhead due to Razor is fairly minimal. Razor-based systems in [3–5, 14] report power overheads that range between 2–8%.

As the supply voltage is scaled, the processor energy (E_{proc}) reduces quadratically with voltage. However, as voltage is scaled below the first failure point (V_{ff}), a significant number of paths fail to meet timing. Hence, the error rate and the recovery energy ($E_{recovery}$) increase exponentially. The processor throughput also reduces due to the increasing error rate because the processor now requires more cycles to complete the instructions. The total processor energy (E_{tot}) shows an optimal point where the rate of change of $E_{recovery}$ and E_{proc} offset each other. Thus, in the context of Razor, a timing error is not a catastrophic failure but a trade-off between the quadratic energy savings due to voltage scaling versus the overhead of recovery due to errors.

The concept of error-detection and correction has traditionally been widely employed in communications and signal-processing applications. In such applications, it is well known to trade-off transmitter power for heavyweight error-correction at the receiver end. In Razor, the concept of error-detection and correction are applied to general-purpose computing.

The RazorI approach relies upon temporal redundancy for timing-error detection. Robustness to Single-Event Upset failures through temporal redundancy has been shown to be particularly effective in a technique first pioneered in [15]. In the RazorI scheme (shown in Fig. 5), every rising-edge triggered critical-path flip-flop is augmented with a so-called shadow latch that samples at the falling edge of the clock. An error signal is flagged when the early speculative sample differs from the correct sample at the shadow latch. In the event of a timing error, a pipeline "restore" signal overwrites the potentially incorrect data in the main flip-flop with correct data from the shadow latch, thereby restoring correct state with a single cycle penalty.

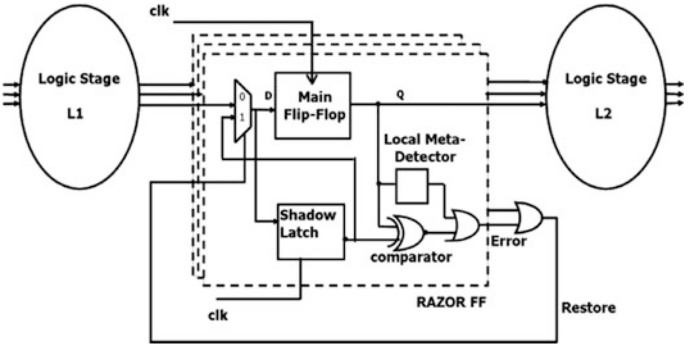


Fig. 5 RazorI flip-flop—conceptual design

The scheme relies upon a pipeline recovery mechanism based on the counter-flow microarchitecture design and requires specialized circuitry for metastability detection and recovery. Process-variability and margining requirements on the metastability detector complicate its deployment in high-performance microprocessors.

The RazorII approach eliminates the need for such a detector by splitting error-detection and correction between circuits and microarchitecture domains. Error-detection occurs exclusively in the RazorII flip-flop and recovery relies upon a conventional check-pointing and replay mechanism that is typically used in most high-performance microprocessors in order to support speculation mechanisms. The schematic for the RazorII flip-flop using a transition-detector is shown in Fig. 6.

Error-detection in the RazorII approach uses a transition detector to generate a pulse out of a transition on the data input. This pulse is then captured within an error-detection window to flag a timing error. The RazorII approach was integrated within an ARM processor implementing a subset of the ARM instruction set architecture. The pipeline design is shown in Fig. 7. Every critical-path endpoint is protected using Razor flip-flops (RFF). The error signal of individual flip-flops is combined together to create the pipeline error signal. The pipeline error signal engages a replay mechanism that restores correct state within the pipeline (Fig. 7).

A Razor dynamic voltage controller compensates for variations by monitoring error rates within the system and adjusting the supply voltage accordingly. The

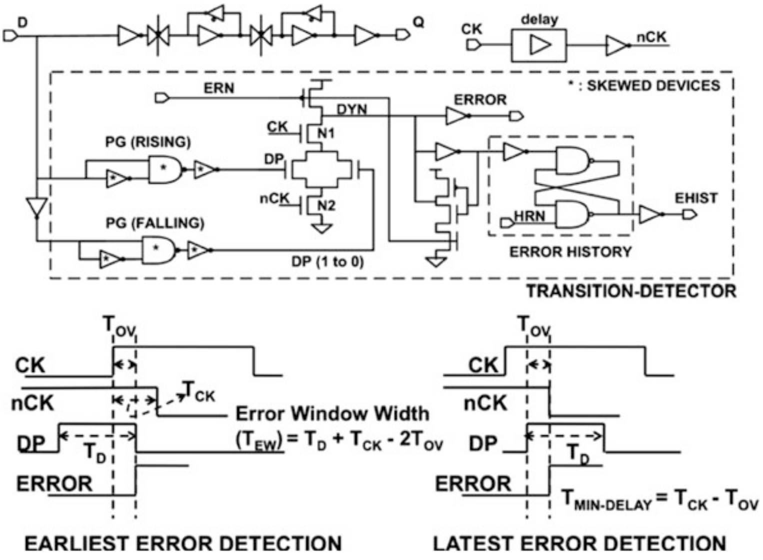


Fig. 6 Transition-detector circuit schematic

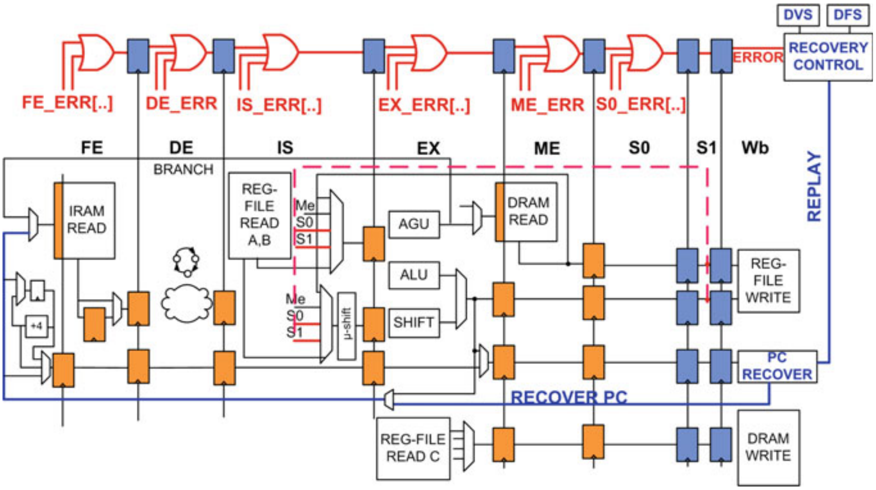
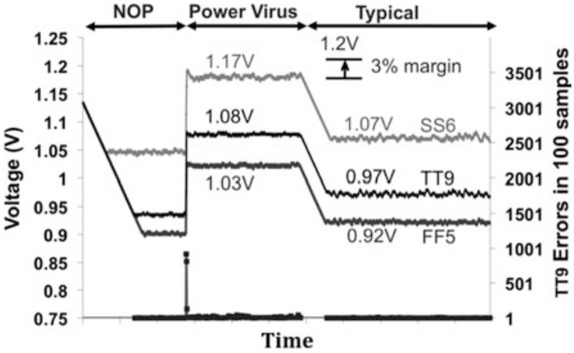


Fig. 7 Pipeline design of an error-resilient microarchitecture

Fig. 8 Response of the razor DVS controller—supply voltage is modulated depending upon the code being executed in the processor. The controller seeks the point of first failure for each code phase



voltage controller automatically tunes the system to operate at the PoFF of the system. The Razor voltage controller response for a code with three distinct phases is shown in Fig. 8 where a 30% energy saving is obtained on a per-die basis by automatically eliminating margins through error-detection and correction.

The Razor approach incurs verification and validation challenges due to the complexities of embedding fine-grained error-detection and recovery within the processing pipeline. On the other hand, data-processing systems such as DSP accelerators are particularly suitable towards error-detection and correction since a data-path dominated pipeline with simplified control typically characterizes them.

In the following, we illustrate how Razor concepts are equally applicable towards DSP accelerators.

5 Error-Resilient DSP Accelerators

Modern mobile and multimedia System-on-Chip (SoC) designs are rapidly evolving into complex, heterogeneous systems. In addition to high-performance application processors, such SoCs rely upon dedicated accelerators to deliver high-performance under stringent power budgets. Unlike microprocessors, DSP accelerators are data-path-dominated with relatively simplified control-plane logic. Such applications are often dominated by tight loops processing large amounts of streaming data, so it is natural to implement these loops as hardware Loop Accelerators (LA). Hardware LAs favourably trade-off surplus transistors to deliver order-of-magnitude higher efficiency compared to the software-only solution in programmable processors, although at the expense of limited or no flexibility.

In [16], we describe the first application of Razor to hardware loop accelerators (RZLA). In contrast with microprocessors, LAs are a class of coprocessors that accelerate a particular function and as such do not need to maintain an internal architectural state. Instead, queues are used in a dataflow-like manner to transfer transient data between functional units. This makes the LAs extremely amenable for implementing Razor recovery, as simply extending existing queues provide the necessary storage for the speculative state in flight, until it is validated using Razor.

Das et al. [16] shows the baseline microarchitecture of the RZLA (Fig. 9). The RZLA is a hardware realization of a modulo scheduled loop. Modulo scheduling is a software pipelining technique that achieves high parallelism by overlapping successive iterations of a loop. The RZLA microarchitecture exploits this parallelism obtained using modulo scheduling through the use of multiple functional units (FUs), each dedicated to a specific operation in the loop. The FUs are labelled ADD (adder), MULT (multiplier), BR (branch unit) and MEM (memory access unit). Unlike microprocessors, the RZLA does not require explicit support for mechanisms such as exception handling. Therefore, state is primarily maintained in Shift-Register Files (SRF) to be consumed when required and then immediately discarded. Wires from the SRF back to the FU inputs allow data transfer from producer to consumer Fig. 9.

The RZLA is architected such that exact recovery is achieved in the event of a timing error. However, most DSP algorithms allow inexactness in the final computational output as long as the algorithmic performance metrics are met. These metrics could be stop-band attenuation in a Finite Impulse Response (FIR) filtre or Peak Signal-to-Noise Ratio (PSNR) in an image compression algorithm. We take advantage of this in wherein we rely upon an approximate error-correction (AEC) algorithm in conjunction with controlled time-borrowing to achieve 37% energy saving in a FIR pipeline [17, 21].

Figure 10 shows the pipeline diagram of 16-tap Razor FIR filter. This design utilizes RFFs at critical-path endpoints to monitor for timing errors. Similar to the RazorII design, the RFF uses a pulsed-latch architecture that is transparent in the high-phase of the clock. Error-detection is postponed to the negative clock-phase. This allows late-arriving transitions to opportunistically time-borrow from the

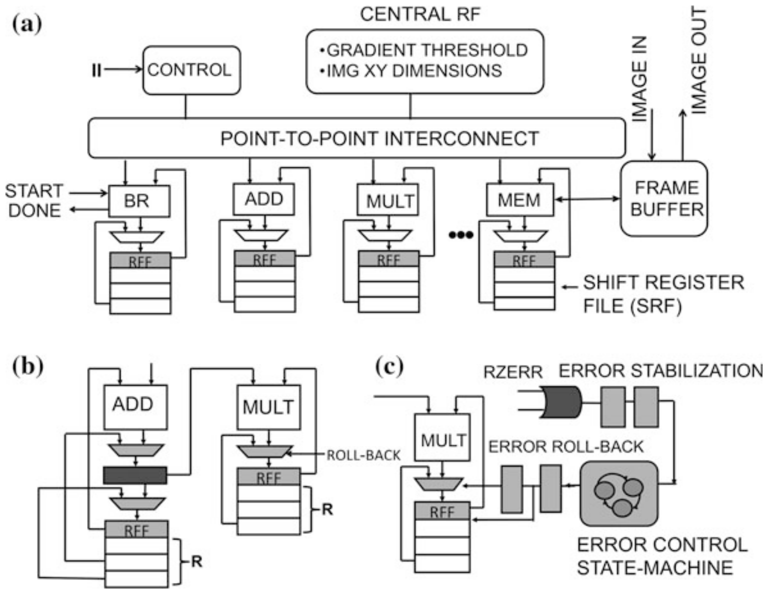


Fig. 9 **a** Architecture of the loop-accelerator implementing the inner kernel of sobel edge-detection algorithm. **b** Extension of shift-register files to incorporate error-correction by keeping **c** Error-control state machine

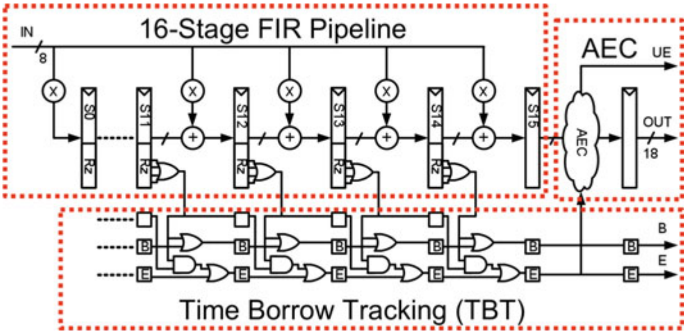


Fig. 10 FIR pipeline implementing a combination of exact and approximate error-correction. Each stage of the FIR pipeline has critical-paths that are protected using pulsed-latch-based razor flip-flops that correct timing errors on the fly. Approximate error-correction is deployed when successive cycles of timing-borrowing is detected

succeeding clock-cycle. Automatic time-borrowing achieves two critical objectives: (1) it leads to a high-performance design enabling 1 GHz operation and (2) it enables on-the-fly error-correction in the event of a timing error.

Thus, error recovery in our scheme is exact in the event of rare timing errors. However, if the error-rate pattern is bursty, it leads to multiple cycles of successive time-borrowing that can potentially exhaust available timing margin. In order to limit the error-magnitude induced due to excessive time-borrowing, we implement an approximate error-recovery scheme that augments exact recovery through time-borrowing. In this scheme, we track successive cycles of time-borrowing. When two cycles of time-borrowing is detected, the AEC block is engaged that replaces the final computational output with a spline-interpolated estimation.

The AEC algorithm uses four neighbouring correct samples (two backward and two forward samples) to generate an estimate of the erroneous sample. Consequently, the algorithmic performance of AEC is impacted as the number of available correct samples reduces. However, the performance with AEC is still significantly better than no error-correction at all.

6 Adaptive Clocking for Supply-Voltage Variations

Typically, error-resilient techniques are robust against all sources of variations. However, they add significant computational resources to implement detection and correction. Several architectural, algorithmic and circuit techniques have to be undertaken in order to limit the resulting overheads of error-correction. Alternative techniques have been pursued to address the overheads and computational complexity of error-resilient techniques. Tracking circuits, ageing monitors and process-binning are examples of such techniques that are comparatively simple in design and implementation. However, these techniques are ineffectual against fast-changing variations such as supply-voltage fluctuations.

Supply-voltage variations are one of the strongest determinants of guardbands in design due to strong correlation of transistor propagation delay with supply voltage. Adaptive-clocking techniques have been developed to particularly address the effect of supply-voltage variations. The key idea in adaptive clocking is to stretch the system-clock frequency in response to supply-voltage variations. Thus, the system clock slows down during periods of supply-voltage droops and increases again when the supply-voltage rises again.

Before we discuss adaptive-clocking techniques, it is important to understand the frequency- and time-domain behaviour of on-chip power-supply networks.

6.1 *Power-Delivery Network Basics*

Figure 11 [21] shows a simplistic representation of the power-delivery network (PDN) composed of a die-package-PCB system [18]. The switching transistors on the die are lumped together and modelled as a current source, I_{DIE}. Explicit on-die decoupling capacitors and non-switching transistors act as local charge reservoirs that are modelled by a capacitor, C_{DIE}. The power-line traces on the package and board are represented using R-L networks. Discrete decoupling capacitors

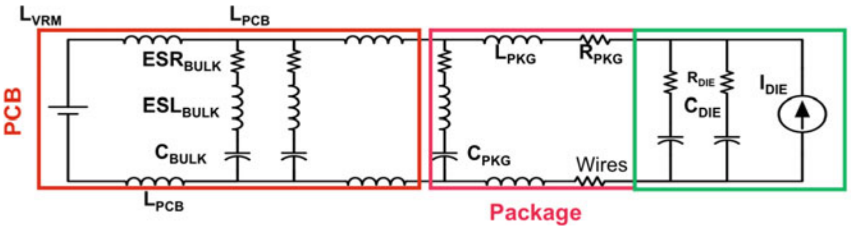


Fig. 11 Simplified representation of a power-supply network

(henceforth, referred to as decaps) on the package (C_{PKG}) and the bulk capacitors on the PCB (C_{BULK}) are modelled by capacitors in series with their effective series resistance (ESR) and inductance (ESL).

$$\Delta V_{\mathrm{DIE}}(t) \cong 2\mathrm{I}_{\max} R + \mathrm{I}_{\max}\sqrt{\frac{2L_{\mathrm{PKG}}}{C_{\mathrm{DIE}}}} \cdot \mathrm{e}^{-\frac{R}{2L_{\mathrm{PKG}}}t} \sin(\omega r - \theta) \tag{1}$$

Equation 1 shows the analytical solution for the voltage droop seen at the die supply rails for such a simplified model of the PDN. The voltage droop can be decomposed into a DC IR-drop term and an AC Ldi/dt term. The resistive component of the droop is addressed by increasing the metallization resources in the PDN. The inductive component is a complex trade-off between the package and the die and far exceeds the resistive droop magnitude in modern computing systems.

Figure 12 shows the PDN input impedance (as seen from the die) as a function of frequency for the simplified PDN in Fig. 1. The impedance spectrum shows three distinct impedance peaks due to each capacitor resonating with its counterpart inductor. The highest impedance peak, referred to as the *first-order resonance,* also occurs at the highest frequency (~ 100 MHz) and is due to the resonance between the die capacitance and the package inductance. The *second-* and *third-order* resonances are due to downstream capacitor networks, and occur at relatively lower frequencies (~ 1 MHz and ~ 10 kHz for the 2nd and 3rd-order resonances, respectively) Fig. 12.

Microarchitectural events such as pipeline interlocks cause current-step excitations that exercise the three prominent system resonance frequencies in the PDN (Fig. 12). The maximum magnitude of the voltage droop is caused due to the first-order resonance, which as such dominates the total timing margin.

6.2 *Adaptive-Clocking Approaches*

Adaptive-clocking techniques slow-down the system clock to mitigate the effect of the first-order supply droop. There are two major categories of adaptive supply techniques. The so-called "analogue" approach provides a continuous modulation

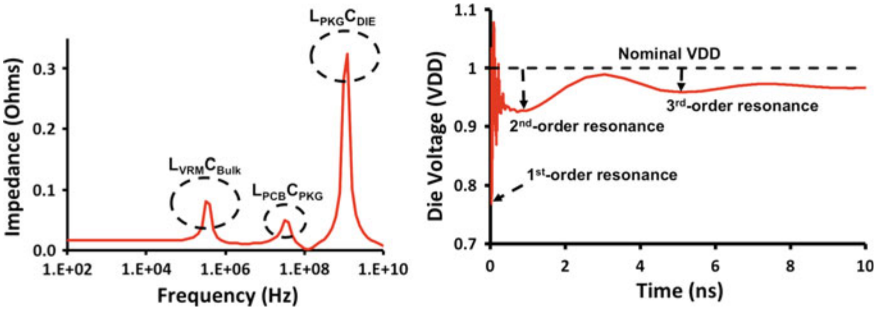


Fig. 12 Frequency- and time-domain response of a Power-Delivery Network (PDN) to a step-current excitation. The PDN response shows the presence of multiple resonance frequencies in the frequency-domain. The time-domain response also shows the same frequencies. In particular, the first-order resonance shows the highest supply-voltage droop at the highest frequency [21].

of the system clock in response to supply-voltage droops. “Digital” techniques, on the other hand, provide thresholded clock-adaptation, i.e. the supply voltage is compared against a certain threshold and modulation is undertaken only when the supply-voltage droops below the threshold.

Kurd et al. describe the analogue approach in [19] wherein power-supply noise is directly mixed into the voltage-controlled oscillator (VCO) of a PLL. The low-bandwidth of the PLL filtres out the high-frequency modulation of the VCO frequency, thereby mitigating concerns against loop stability.

Grenat et al. [20] present a “digital” modulation technique wherein the supply voltage is compared against a programmable threshold. A threshold-crossing initiates a clock-modulation scheme wherein cycle stretching is achieved by choosing successive phases out of the output of a Delay-Locked Loop.

Thus, adaptive clocking enables elimination of a subset of guardbands due to high-frequency supply-voltage fluctuations. Nominal operation of the system occurs at a higher frequency and the system slows down only when a droop-condition is encountered. Adaptive clocking is limited by the response latency that is limited by the clock tree depth and timing delay incurred in detection and initiated response in the event of a voltage droop [22]. Ensuring robustness when operating under reduced guardbands is a key challenge for adaptive technique approaches.

7 Conclusion

In this chapter, we reviewed various technological challenges for variation-tolerant computing. We reviewed three classes of techniques, namely tracking circuits, error-resilient computing and adaptive clocking. In particular, we focused on a particular flavour of error-resilient technique called Razor that enables energy-efficient operation by actively allowing timing errors to occur. We reviewed various approaches applied to university and industrial processors that demonstrate reliable energy-efficient operation using Razor-based dynamic adaptation. We showed measurement results where Razor error-correction enables robust operation in the presence of radiation-induced SER failures. Variation-tolerance remains a key design challenge, particularly as process technology scales to sub-10 nm critical dimensions. None of the techniques described in the chapter are perfect silver bullets due to the trade-offs between complexity, efficiency and engineering applicability that are involved. Hence, there is an urgent requirement for continued research investment in this area, both in academia and in industry. As process technology reaches fundamental physical limits, such techniques will prove to be an effective recourse to reliable computation in presence of failure-prone transistors.

References

1. H. Esmaeilzadeh et al., Dark silicon and the end of multicore scaling. Micro IEEE **32**(3), 122, 134 (2012)
2. D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N.S. Kim, K. Flautner, Razor: circuit-level correction of timing errors for low-power operation. IEEE Micro **24**(6), 10–20 (2004)
3. S. Das et al., A self-tuning DVS processor using delay-error detection and correction. J. Solid-State Circ. (2006)
4. S. Das et al., RazorII: in situ error detection and correction for PVT and SER tolerance. IEEE J. Solid-State Circ. **44**(1), 32–48 (2009)
5. D. Bull, S. Das, K. Shivashankar, G. Dasika, K. Flautner, D. Blaauw, A power-efficient 32 bit arm processor using timing-error detection and correction for transient-error tolerance and adaptation to PVT variation. IEEE J. Solid-State Circ. **46**(1), 18–31 (2011)
6. J. Tschanz et al., Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging, in *2007 IEEE International Solid-State Circuit Conference* (2007), pp. 292–293
7. K.J. Nowka et al., A 32-bit POWERPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling. IEEE J. Solid-State Circ. **37**(11), 1441–1447 (2002)
8. A. Drake et al., A distributed critical-path timing monitor for a 65 nm high-performance microprocessor, in *IEEE International Solid-State Circuit Conference* (February 2007), pp. 398–399
9. T. Fischer et al., "A 90-nm variable frequency clock system for a power-managed itanium architecture processor. IEEE J. Solid-State Circ. 218–228 (2006)
10. R. McGowen et al., Power and temperature control on a 90-nm itanium family processor. *IEEE J. Solid-State Circ.* 229–237 (2006)
11. S. Das D. Blaauw, Adaptive design for nanometer technology, in *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009* (2009), pp. 77–80
12. S. Youngmin et al., 28 nm high-metal-gate heterogeneous quad-core CPUs for high-performance and energy-efficient mobile application processor, in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers* (*ISSCC*) (17–21 February 2013), pp. 154, 155
13. F. Masaki et al., A 28 nm high κ-metal-gate single-chip communications processor with 1.5 GHz dual-core application processor and LTE/HSPA + -capable baseband processor, in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers* (*ISSCC*) (17–21 February 2013), pp. 156, 157

14. K. Bowman et al., A 45 nm resilient microprocessor core for dynamic variation tolerance. IEEE J. Solid-State Cir. **46**(1), 194–208 (2010)
15. M. Nicolaidis, Time redundancy based soft-error tolerance to rescue nanometer technologies, in *Proceedings of the IEEE VLSI Test Symposium* (April 1999), pp. 86–94
16. S. Das, G. Dasika, K. Shivashankar, D. Bull, A 1 GHz hardware loop-accelerator with razor-based dynamic adaptation for energy-efficient operation, in *IEEE Custom Integrated Circuits Conference* (September 2013)
17. P. Whatmough, S. Das, D. Bull, A low-power 1 GHz razor FIR accelerator with time-borrow tracking pipeline and approximate error correction in 65 nm CMOS, in *IEEE International Solid-State Circuits Conference* (February 2013), pp. 428–429
18. J. Tschanz et al., Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance, in *2009 Symposium on VLSI Circuits* (2009) pp. 112–113
19. N. Kurd et al., A Family of 32 nm IA processors. IEEE J. Solid-State Circ. **46** (1), 119–130 (2011)
20. A. Grenat, S. Pant, R. Rachala, S. Naffziger, Adaptive clocking system for improved power efficiency in a 28 nm x86-64 microprocessor, in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers* (*ISSCC*) pp. 106–107
21. S. Das, P.N. Whatmough, D. Bull, Modeling and characterization of the system-level power delivery network for a dual-core ARM cortex-A57 cluster in 28 nm CMOS. ISLPED (2015)
22. P.N. Whatmough, S. Das, D. Bull, Analysis of adaptive clocking technique for resonant supply voltage noise mitigation. ISLPED (2015)
23. M. Gupta et al., Cross-layer system resilience at affordable power, in *2014 IEEE International Reliability Physics Symposium* (June 2014)
24. P.N. Whatmough, S. Das, S.D.M. Bull, I. Darwazeh, Circuit-level timing error tolerance for low-power DSP filters and transforms. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on **21**(6), 989–999 (2013)

Design for Test and Test Equipment Roadmap

Davide Appello

Abstract This chapter discusses the expected evolution of testability and test equipment capabilities and performances for ICs manufactured on VLSI technologies. A review of the evolution of failures modes across the technology platform is used to evaluate impacts on the requirements for testability and test equipment (ATE). These same elements are also evaluated toward the quality and reliability requirements offered by different market segments. Finally, considerations on costs and economical sustainability on the effect of requirements are reported to the readers.

1 Introduction

If in the early years of VLSI, the testing of ICs were essentially solved through functional test, the 90's and 00's years saw the structural test approach through scan and ATPG or BIST as the dominant test paradigm. New technology nodes are offering several different reasons to force changing, or at least upgrading that paradigm.

The scaling of technologies occurred in the last decades determined an increase of complexity which put significant challenges to the development of adequate EDA solutions capable to develop effective and efficient testing. Yield management tools and diagnostics played a big role in allowing understanding and then improving yield. A new key factor is the progressive gap which new technologies are putting in the effective adherence of fault models to the actual defects present in IC.

D. Appello (✉)
STMicroelectronics S.R.L, via C. Olivetti, 2, 20864 Agrate Brianza, MB, Italy
e-mail: davide.appello@st.com

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_8

This creates major challenges to the DfT and in general to the test technologies.

- ATPG based on fault models like stuck-at, transition and bridging is now giving reduced confidence to be exhaustive.
- Devices are operating at always lower voltages and high leakages and consumption are complicating the modeling of the device behavior, because of
  - Increased sensitivity to variations,
  - Exasperate operating conditions (very high temperature, severe mission profiles),
  - Exasperate quality and reliability requirements for products targeted to medical, automotive and avionics domains, for example.

The concerned chapter will offer some perspective and analysis on how these challenges can be faced and hopefully resolved.

New synergies between DfT, test equipment, and test methods shall be proposed to highlight cause-effects relations. Special attention shall also be given to the sustainability of the costs of the proposed solution.

The rest of the chapter is organized as follows: the first section will give analytical details through examples of the effects of parametric variations. The following section will instead focus on the adherence of currently utilized fault models with respect to actual defects. Immediately after, we will bring to the attention on the fault activation limitations offered by traditional design for test practices, such as scan. In the same section the consequences on test time are discussed. The fourth section is focused on the testing hardware, reporting the state of the art and indicating the key obstacles in place respect the new needs. The final and fifth section will discuss of the new frontiers opened by the presented scenario. In this part of the chapter, some considerations will be taken to both guess possible solutions to satisfy new test paradigm as well highlighting the new challenges put on the testability methods.

2 Parametric Variations

The goal of this section is giving the reader information regarding the effects on the testing process put by parametric variations (PV) [1, 2]. PV's typical and most undesired effect is to make device behavior less homogeneous respect a set given electrical stimuli. For example, a test paradigm in place since decades supposed the existence of a deterministic relation between a set of functional vectors given in input to a circuit and the responses of the same. Another paradigm which gained progressive consensus since the 90s was the systematic adoption of structural test methods to provide test coverage at manufacturing test. The performances and quality offered by ATPG tools, beside relatively simplicity of use, facilitated the usage structural test as the main instrument to achieve desired test coverage, leaving behind functional test methods.

These conditions progressively became less valid by accessing newer and smaller technologies. As a matter of fact, it is nowadays no more realistic pretending that a set of test vectors is able extending its applicability to any parts manufactured across the entire process window. This situation is emphasized in the case other dimensions like voltage and temperature are considered. With the increase of complexity, the activation conditions of structural test progressively diverged respect those achievable in functional mode. The activation of nonfunctional paths and the higher switch mode activities are the main contributors to the enlargement of the gap together with the increased leakage present with finer technologies (see also discussion in chapter "Application Scenarios").

Several mitigation methods have been tried to reduce the effects of PV. In some cases, the objective is to reduce the gap with functional mode. Low-power ATPG aims achieving this result, by reducing the switching activity. In this case, the main effect is the inflation of vector count and hence, test time.

Other methods are making use of adaptive test methods. The main goal is to structure test in two separate phases:

- The first phase applies test whose goal is to appreciate the relative performance of the device under test (e.g., its positioning in the process window)
- The second phase applies vectors which are parametrized on the results of first phase

The first phase makes sometime use of specific vectors sets signed-off in different conditions. In other cases [3] process monitors are deployed in relevant device location to appreciate relative performances.

In addition, other methods known as RBB/FBB (Revers or forward body bias) aims controlling relative threshold acting on biasing of device body.

3 Fault Modeling

The industry is using fault models since at least three decades. A fault models aims modeling a defect in the hardware is such a way that a simulator can identify locations in the circuit model compatible with the occurrence of faults of the identified type. In a second phase, a set of stimuli can be generated with various techniques to activate those faults and propagate their effects to suitable observation points. Stuck-at faults have been widely utilized since those times and progressively increasing coverage targets toward and above 99%. This metric remains a typical achievement milestone in designs nowadays. However, the understanding is that its achievement ensures that the design is well controllable but not that there is the ability of observing effects of defects whose behavior is not relatively close to an actual connection between a wire and the supply or ground rail. Most important is that there is no metric respect non DC failures type. Starting from this consideration, additional fault models have been considered. Some of these aimed targeting specific defects mechanisms like bridges, but unfortunately falling into similar limiting aspects for less-than-stuck behavior. IDDq had some success and merit in modeling the analog behavior of defects also. Unfortunately, the increase of sub-threshold leakages of small lithography technologies made its utilization impractical if not ineffective.

Transition delay faults (TDF) have been introduced since almost two decades to offer broadside coverage of transition delays. The achievement of very high coverage figures is not straightforward. Moreover, ATPG algorithms tends to stop generating patterns as soon as a transition on a net is activated independently if to do the most critical path was not used. This limits its effectiveness with respect to the ability to effectively appreciating the actual performances of a circuit in presence of defectivity and/or variations. Their effectiveness may probably be limited to the detection of stuck-open faults [4] To go beyond this limitation, some algorithm evolution is actually linking the path criticality information extracted from design with the ATPG. In this way, the ATPG keeps generating vectors until a transition is generated using the most critical path. By doing this, inflation to the overall pattern count is usually visible, impacting the test application time along with the test data volume required. Other approaches to improve modeling of faults included bridge faults and cell-aware defects. In the first case, analysis of critical layout configuration with respect to target defect is performed to drop concerned nets into a fault dictionary for the testing [5]. In cell-aware approach, libraries of cells shall be analyzed to identify critical layout configuration for opens resistive bridges [6]. In analog design, specific considerations shall be taken as in [7].

4 Faults Activation and Impact on Test Time

Scan-based testing efficiency in reaching high coverage has some drawbacks, which progressively become more evident with the shrinking of technologies. The relatively high switching activity compared with functional mode produces consumption and voltage droop conditions which may determine fails. This effect is emphasized by the progressive reduction of operative voltage window respect the available process window. The presence of localized if not global voltage drop due to consumption may drive the device working in marginal conditions, eventually creating failures during testing. Below chart shows an example of a characterization exercise performed on a representative sample set. From top to right, results distributions from the same sequences applied to material taken from different corners in the available process window are reported. From left to right, the first column report results at −45 °C, the central column at 25 °C the last column on the right shows results of tests performed at 150 °C.

The reader can easily observe how it is hardly identifiable a unique set of condition which accommodates for any material, eventually for a specific temperature (Fig. 1).

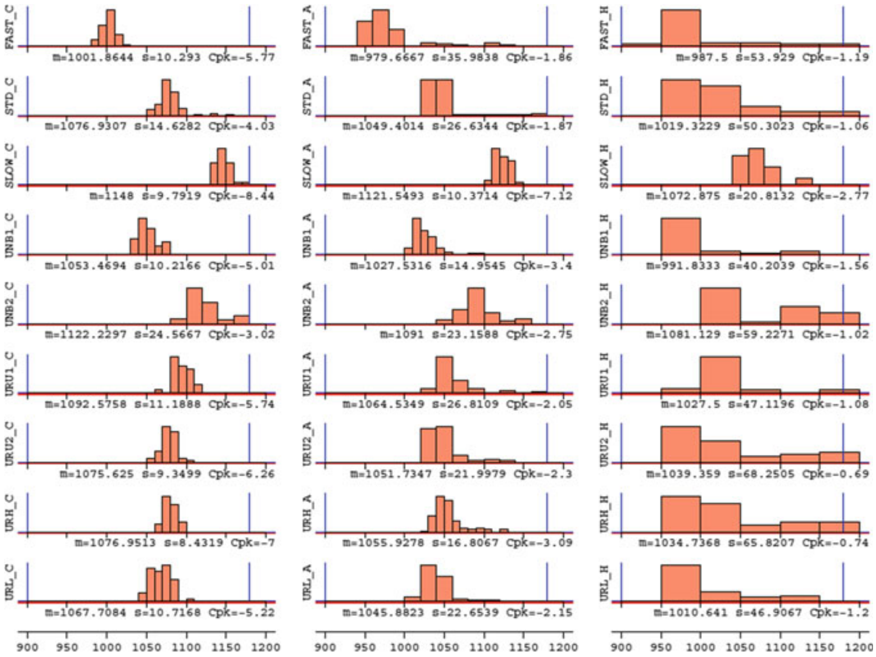


Fig. 1 The plot reports chraterization results of corner lots, measured in various voltage and temperature conditions

Possible activities to mitigate these effects essentially regard test adaptation. In combination of adaptive test method, infrastructure-IPs can be utilized [3] to assess in deterministic way SoC performances and accordingly to that set most adequate test conditions and test limits.

5 Test Hardware

The reduced margins of device operations and consequent need of increased test accuracy exasperate the focus on test hardware to maximize performances. There are at least three categories of limiting factors in delivering test performances to the device. The first category is related to the interconnecting hardware like probe cards and sockets. In both cases, they represent a variation with respect to the impedance conditions that the IC will see in the actual application board where the device is soldered. These variations are mostly due to contact resistance and capacity of pins/needles. The determined impedance network puts limitation to the achievable performances. Series resistance added to the supply line modifies the actual voltage delivered to the die. For high-speed interfaces, limiting the analysis to testing of packaged devices, the effects of socket and test board is usually to severely limit the achievement of targeted performances.

On top of intrinsic performances offered by above described tools, the reader shall know that package level test usually suffers of other negatively influencing factors. These are due to accidental variations due to debris particles of materials which may aggravate the impedance performances. The same effect is determined by other environmental conditions of testing at high temperature and especially during the test at low temperature (typically −45 °C and below).

Another perspective that shall be considered concerns test time. In the previous section, it have been highlighted a convergent effect of multiple factors (variations, noise due to test activation) determining exponential increase of test time (from few seconds to several minutes). The most conventional paradigm for package test foresees the utilization of automatic test equipment (ATE) and IC handlers. For high complexity and high lead/ball count packages, the most utilized handlers adopt a principle called “pick and place”. Increasing test parallelism (also said multi-site test) is widely recognized as an effective approach to reduce unitary cost of test. Unfortunately, the handling technology is showing an asymptotic behavior also captured from ITRS roadmap [8]. As consequence, in case no countermeasures will be taken, an increase in the cost of testing will likely be observed. In the next section, we will try drawing some conclusions regarding possible solutions to the problems described especially in Sect. 3 and within this section.

6 Challenges and Opportunities

From the previous section, it emerged the concurrence of three main factors suggesting for new test methods.

1. Reduced consistency between fault models and actual defects
2. The environmental conditions offered by traditional ATE/Handler/tooling makes difficult achieving targeted device performances
3. Test time increase.

Starting from the third factor, it appears evident the requirement of significantly increase test parallelism much beyond the capabilities indicated by the ITRS roadmap [8].

This is requiring a shift of handling technologies toward test board loading made off-line respect test. This approach is widely utilized in other manufacturing systems such as for the burn-in and also for system-level test [9].

To face the first factor, trends in the industry are suggesting combining structural test with more functional activation. This represent an inversion of trend respect the past 20 years, but underline how the need of exploring operative corners (voltage and temperatures) which can hardly be reached with structural tests because of very likely yield overkill situation. One challenge will be to define suitable metric for this approach to effectively correlate coverage with quality achievements.

The second of the points reported is and remains an intrinsic problem related to the testing of packaged devices. Contacting technology will never equal the

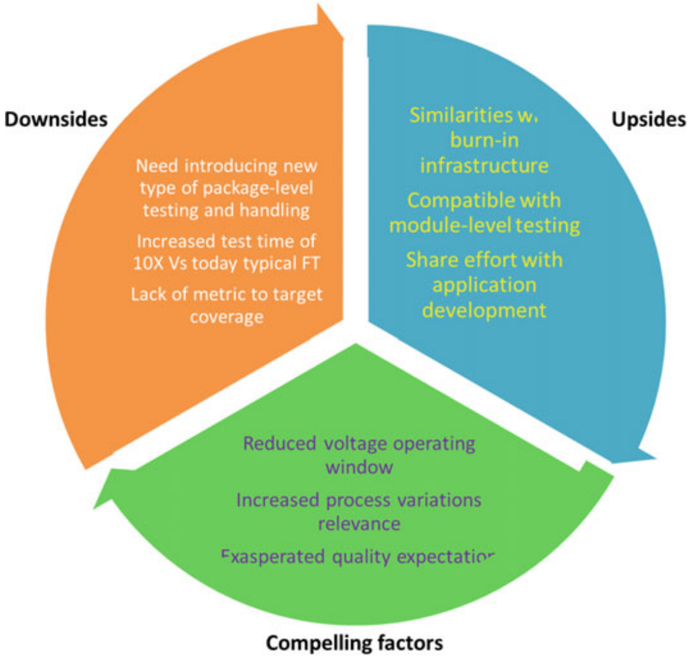

Downsides
Need introducing new type of package-level testing and handling
Increased test time of 10X Vs today typical FT
Lack of metric to target coverage
Upsides
Similarities w. burn-in infrastructure
Compatible with module-level testing
Share effort with application development
Reduced voltage operating window
Increased process variations relevance
Exasperated quality expectatio
Compelling factors

Fig. 2 Challenges and opportunities chart

contacting impedance of a device soldered to its application footprint. Indeed, another trend in the industry shows that testing at module-level might become a suitable approach to face the problem. Figure 2 summarizes the motivations, the challenges and the opportunities for the next generation testing.

As a summary, the above considerations lead to some hypothesis on the future needs and challenges.

ATPG and other structural test. The industry will continue to rely on structural test as main method to screen manufacturing defects since they provide analytical coverage metrics and independence from device functionality on analytical metrics. However, methods to reduce power during test w/o impacting test duration are highly desirable.

Functional test. Test methods in this area are necessary to close the coverage gap opened by limited applicability of structural test across the entire window (process, voltage, temperature) [10]. Development of metrics is highly desirable.

Stress test methods. The paradigm used by burn-into screen early-life failures methods shall be reviewed, in light of condition put by new technologies (finfet, SiGe) and by high power demanded by SoC processors.

Packages. The exponential increased complexity of packages (adoption of bumps/pillars and flip-chip) will put noticeable challenges on the qualification process for critical mission profile products like on the screening of assembly-related defects.

References

1. R. R. Rajeev, D. Blaauw, D. Sylvester, A. Devgan, Modeling and Analysis of Parametric Yield under Power and Performance Constraints. in *IEEE Design & Test of Computers*, (Ann Arbor, July–August 2005)
2. I. Polian et al., Towards Variation-Aware Test Methods, in *IEEE, Sixteenth IEEE European Test Symposium*, 2011
3. Y. Zorian, What is an Infrastructure IP? in *IEEE Design & Test of Computers*, vol. 19, no. 3, (May–June, 2002), pp. 5–7
4. A. D. Singh, Scan Based Two-Pattern Tests: Should They Target Opens Instead of TDFs? in IEEE *16th Latin-American Test Symposium* (LATS),(Auburn University, AL, 2015)
5. F. J. Ferguson, T. Larrabee, Test pattern generation for realistic Bridge fault in CMOS ICs, in *Proceedings of IEEE Int'l Test Conference, ITC, 1991*, (1991), pp. 492–499
6. F. Hapke, M. Reese, J. Rivers, A. Over, V. Ravikumar, W. Redemund, A. Glowatz, J. Schloeffel and J. Rajski, Cell-aware Production test results from a 32-nm notebook processor, in *Proceedings 2012 IEEE International Test Conference*, (MentorGraphics)
7. B. Kruseman et al., Defect Oriented Testing for Analog/Mixed-Signal Devices, in *IEEE International Test Conference*, 2011
8. ITRS Roadmap for Semiconductors-Test Chapter, 2013, www.itrs2.net
9. S. Biswas, B. Cory, An Industrial Study of System-Level Test, Copublished by the IEEE CEDA, IEEE CASS, and IEEE SSCS, IEEE D&T Magazine, (2012)
10. H. H. Chen et al., Predicting System-Level Test and In-Field Customer Failures Using Data Mining, in *IEEE International Test Conference*, 2013

Resilience Proportionality—A Paradigm for Efficient and Reliable System Design

Vilas Sridharan and Sudhanva Gurumurthi

Abstract Reliability, Availability, and Serviceability (RAS) are key considerations in hardware design, be it for mobile devices or high-end servers. However, provisioning RAS is often at odds with meeting performance and energy targets and increases the overall cost of design of the chip. As a result of this tension, chip design companies have to make difficult decisions about how much RAS they can incorporate into each product in their portfolio and even what customers and market segments they can realistically target. On the other hand, highly scaled silicon technology nodes are susceptible to a variety of reliability problems and emerging technologies such as die-stacking and non-volatile memory, while critical for meeting the demands of future computing needs, have significant reliability challenges of their own. RAS features can actually serve to reduce the deployment costs of these technologies (e.g., by increasing effective yield). Determining the tradeoff between design cost, deployment cost, and the RAS needs of a market is the critical issue to address when evaluating RAS features. In this article, we shed light on this struggle between driving greater efficiency, lowering costs, and meeting the RAS demands of various market segments from an industry perspective. We argue that ending this struggle requires having sufficient flexibility in the design to adapt to the needs of a wide range of applications and hardware configurations. We call such an approach "resilience proportionality" and believe that this approach should guide future architectural reliability research. Finally, we discuss how resilience proportionality can be achieved and certain challenges that need to be addressed.

V. Sridharan (✉)
RAS Architecture, Advanced Micro Devices, Inc, 90 Central St, Boxborough,
MA 01719, USA
e-mail: vilas.sridharan@amd.com

S. Gurumurthi
AMD Research, Advanced Micro Devices, Inc, 7171 Southwest Pkwy, Austin,
TX 78735, USA

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_9

1 Introduction

The microprocessor industry has been at the forefront of driving growth in the computing power of modern systems, from smartphones and tablets to servers used in warehouse-scale computers such as cloud and high-performance computing (HPC) systems [1]. The diverse needs of these computing platforms and applications have driven innovations and diversity in the processing elements themselves, from CPUs that span a large range of power operating points and instruction set architectures (ISAs), graphics processing units (GPUs), accelerated processing units (APUs) that integrate CPU and GPU technology on a single chip, and emerging memory technologies such as die-stacking and non-volatile memory (NVM). While higher performance and lower energy usage play a major role in driving product roadmaps and indeed tend to be the characteristics that are most obvious to the end-user of any computing system, the practical viability of the system also depends on whether it can perform computations correctly and continue doing so over the expected life of the part.

Any computing system, be it a handheld device or a server for a high-performance computing (HPC) system, has a specific reliability requirement [2]. This requirement is typically quantified using a metric called Failures In Time, or FIT. One FIT equals one failure every one billion hours of operation. While the FIT metric appears to define failures over an extremely long timescale, one has to also consider the context in which the part is used. For example, a data center or supercomputer may have tens of thousands of processors operating simultaneously. In this case, the failure rate of the system is the product of the FIT rate of each part and the number of such parts in the system. Therefore, the FIT rate of any one component has to be sufficiently low to satisfy the system-level reliability requirement.

A device with a low FIT target is required to be more reliable than a device with a high FIT target. In general, client systems such as phones, tablets, and laptops tend to have less stringent reliability requirements (i.e., higher FIT rate targets) than server, high-performance computing, or embedded (e.g., automotive) systems [3]. However, there may be specific applications even in the client space that may have more demanding FIT targets.

In order to meet a given FIT target, a device must be designed to be resilient to the reliability problems that occur. Silicon semiconductors are susceptible to a variety of reliability problems, including transient and permanent faults. Transient faults are random bit flips that do not damage the circuit but can corrupt data that is computed or state stored in memory [4]. Errors due to transient faults can be corrected by over-writing the bit(s) with the correct value. Permanent faults, on the other hand, cause an incorrect value to be returned repeatedly and will usually require disabling or replacing the failing component [5].

Memory structures, such as caches and main memory, are especially vulnerable to transient and permanent faults. The prevalence of these faults in production systems has been documented in recent publications [6]. Because memory structures occupy a large fraction of the die area in most chips, protection needs to be added to ensure that these faults do not silently corrupt computation and memory state. Second, emerging technologies, such as die-stacked memory and NVM, which are being actively used by industry for meeting the demands of future computing needs, have significant reliability challenges of their own [7]. For example, many NVM technologies suffer from limited endurance, whereby repeated writes to the memory cells can eventually cause permanent faults in those cells.

There are a variety of industry-standard techniques used to implement reliability in processors and in memory systems. These techniques span the various stages of the design of the part and may involve additional steps by a system integrator or even the end-user. Process nodes may be optimized to make the transistors resilient against faults and circuit-level guard-banding techniques can be used to set operating margins that reduce the likelihood of certain types of failures over the intended lifetime of the part [8]. Radiation hardening is another circuit-level technique that improves resiliency by designing storage elements to be substantially less vulnerable to transient faults [9]. At the microarchitecture-level, parity and error-correcting codes (ECCs) can be used to detect and correct errors [10]. Higher level redundancy techniques such as full processor duplication and lock-stepping may be employed for achieving even higher levels of reliability at the system-level [11]. Finally, system-level error detection and recovery techniques may further enhance resiliency [12].

As may be evident from this description, reliability comes at a cost. The question for chip vendors is how much cost is worthwhile. The next sections delve into this issue.

2 Providing Reliability: A Market View

From a product perspective, not meeting a reliability requirement can mean lost opportunity in terms of capturing a certain market or customer, loss of customer satisfaction, or worse, a faulty or unsafe product that may lead to product recalls! For example, around the year 2000, Sun Microsystems' flagship enterprise server line suffered from unpredictable crashes due to insufficient protection in the caches and a major telecommunications customer stated it would switch from Sun to a competitor for its next batch of servers [13]. Another less obvious consequence of insufficient reliability is that high rates of service calls due to reliability problems in the field may cost a company hundreds of person-years (and potentially millions of dollars) to diagnose and debug. For example, a processor can experience a super-linear increase in failure rate with only a small increase in structure size [14]. If this type of phenomenon were first observed in the field, it could take months to diagnose and root-cause.

Due to these potential consequences of insufficient reliability, companies need to weigh the performance, power, and die area costs of providing RAS against the opportunities for growth and profitability in those markets that require the protection.

3 The Dilemma of Providing Reliability

From the point of view of a chip designer, reliability has an *opportunity cost*: adding reliability features to a product may take away the opportunity to optimize some other aspect of the design. For example, techniques such as parity and ECC consume die area that might have otherwise been used to boost performance, energy efficiency, or add other functionality to a design. Many reliability mechanisms also entail performance and power overheads that add to the overall design cost [15].

By and large, this cost can be quantified as a dollar cost to the gross margin of a company (sales revenue minus cost of goods sold), either due to increased die area and thus cost per device or lower selling price due to reduced performance or increased power. The overall cost to a company of providing reliability is the gross margin cost per chip times the number of chips shipped. This means that reliability can be most expensive in high-volume markets, even if the gross margin per chip is modest. Many high-volume markets (e.g., phones and tablets) also have low reliability requirements, meaning that the number of errors that need to be handled (e.g., detected, corrected, or avoided) is small. Therefore, adding cost to the chip in order to meet these requirements is extremely inefficient even though *some* reliability hardware is often required. Compare this to high-end markets (e.g., HPC), where volumes are low and reliability requirements are high, so many errors must be avoided, making reliability efficient.

This tradeoff is quantitatively illustrated by the solid line in Fig. 1. The figure shows the cost of ECC-based protection per unit die area across a range of market segments. The data in this graph is based on typical die counts, node and system FIT targets for each market segment, and representative volume shipments for each market. The solid line also assumes a single design that serves all markets. As mentioned previously, node FIT rates are lower than system FIT rates and the node counts for server systems can be large. However, overall volume shipments for client parts tend to be significantly higher than those for servers. Therefore, additional area in a client part has a larger impact to a company's gross margin (and therefore its bottom line) than additional area in a server part, as demonstrated by the solid line in Fig. 1.

As the graph shows, the efficiency of using a certain amount of die area for reliability is lower for the higher-volume markets (e.g., phones or tablets) than those with lower volumes (e.g., servers). Therefore, cost considerations pose greater hurdles to implementing reliability on client and other high volume parts. If volumes remain unchanged, lowering the cost of reliability requires reducing the cost of reliability per unit area, or increasing the FIT target and therefore reducing the reliability of each device. Doing the latter puts in jeopardy the ability to create a viable and competitive product for the target market segment, so finding ways to lower the cost of meeting a given level reliability is the preferred option.

One way to reduce margin costs for a given market is to design different devices for different markets, but this is often impractical due to up-front non-recurring

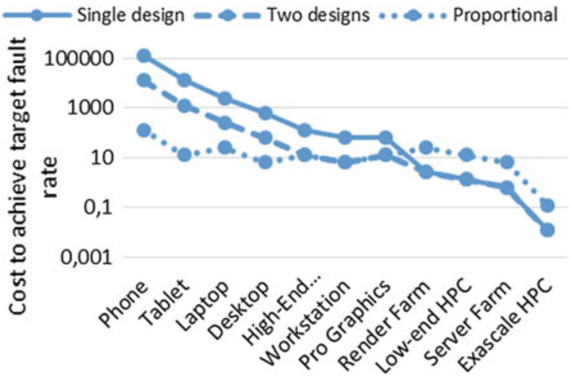


Fig. 1 The cost of RAS. This graph shows the total gross margin cost of providing protection across a range of market segments, from mobile systems to high-end HPC, in three different scenarios. The key point is that providing reliability is efficient at the high end, where targets are stringent and system fault rates are high, but volumes are low, and very inefficient at the low end, where targets are looser, system fault rates are lower, and volumes are high. A key goal of RAS research is to be more efficient at the low end, which reduces the overall cost of providing reliability

expense (NRE) cost to design the chip. As a result, even the largest CPU vendors can only afford to produce one or two CPU designs per generation, and each design is required to scale across many market segments. A scenario where two different CPU designs are used to address client and server markets is shown by the dashed line in Fig. 1. As the figure shows, the CPU design is still inefficient on the low end, despite a reduction in cost of an order of magnitude.

An ideal scenario is to implement RAS such that the cost-per-error is low and is consistent across the spectrum of markets. That is, RAS can be implemented with low overheads for any product and with low design cost regardless of the requirements of the target market segments. This ideal scenario is shown by the dotted line in Fig. 1, which shows a cost reduction of three orders of magnitude at the low end. As the figure shows, the techniques used can actually be less efficient at the high end, but it is still a net win if overall gross margin cost is minimized. In some senses, the dotted line in Fig. 1 can be viewed as one of the few primary goals of RAS research: to “bend the cost curve” and become more efficient, especially in markets at the low-end of the reliability range.

4 Making Informed RAS Decisions to Reduce Costs

The workloads running on the systems can impact their overall error rate and is quantified using a metric known as the architecture vulnerability factor (AVF), which is expressed as a percentage [16]. A low AVF means a low likelihood that a fault will impact the externally visible state of the system (i.e., result in an error) whereas a high AVF means a higher error rate. The datapoints in Fig. 1 assume a fixed AVF of 40%, but we swept the AVF values from 10% up to 80% with similar results. Note that AVF can have an order-of-magnitude impact on the cost-per-error between the lowest and highest values in the AVF range.

AVF analysis can facilitate prioritizing which hardware blocks within the chip need to be protected and what type of protection to incorporate (e.g., detection-only versus detection+correction) to minimize the die area impact, but there are still costs associated with the design cycles for implementing RAS. Furthermore, since these markets may be served by chips whose microarchitectures or even architectures differ (e.g., CPUs versus GPUs), the RAS features have to be designed and customized for each such product, adding to the design costs. Therefore, while AVF analysis is a valuable addition to the RAS architect's toolbox, there is scope for further reduction in the overall cost of implementing reliability.

5 Going One Step Further—Resilience Proportionality

Barroso and Hölzle pointed out that in order to improve the energy-efficiency of computing systems, it is desirable for the energy utilized by the system to vary in proportion to the work done (i.e., computational load) on the system [17]. That is, in any given moment of time, a system that does no work should consume no energy, a system that under heavy load can utilize the maximum amount of energy to perform the work, and in between those extremes, the energy usage should be proportional to the load, preferably consuming the least amount of energy at each operating point. Barroso and Hölzle coined the term *energy proportionality* to refer to this property. Energy proportionality has been a major driver for many changes, from the design of individual components of a server to the design and management of entire warehouse scale computers [12].

The notion of proportionality is a valuable guiding principle for reliability as well. That is, one should be able to provide a high degree of resilience when strong reliability guarantees are required and it should be possible to "dial down" the resilience in proportion to the reliability needs, going all the way down to no protection at all when no reliability guarantees are required. We call this property *resilience proportionality*. Varying the strength of the protection based on reliability needs can allow for more optimal performance and energy behavior. Indeed, resilience proportionality can help achieve energy proportionality.

Realizing resilience proportionality and truly achieving reliability cost savings will require research and development into several areas. First, being able to vary the strength of the protection mechanism will require the development of RAS mechanisms that are flexible and provide a sufficient number of reliability operating points (similar to CPU power states). Such flexible RAS mechanisms can be implemented through hardware, software, or a combination of the two. For any hardware mechanism, care needs to be taken to ensure the die area used by the feature and the design, implementation, and verification complexity do not outweigh the benefits of flexibility. A software-based approach can alleviate the die area and hardware complexity costs, but will entail software design and verification costs and will need to provide guarantees in terms of the detection/correction coverage.

Second, the advent of heterogeneous hardware such as GPUs and other accelerators in the SoC and the computing platform pose a challenge. One would need to develop flexible RAS mechanisms for each component and the set of mechanisms have to be compatible with each other so that one could compose any arbitrary heterogeneous system. Achieving such designs cost-effectively will require innovations in architecture, runtimes, and EDA tools. Software-based approaches could address some of these challenges, if the flexible RAS mechanism can be incorporated transparently into the compilation path of accelerator programming languages such as OpenCL or those of accelerator architectures such as the Heterogeneous Systems Architecture (HSA) [18, 19]. An example of ongoing research in this area is the compiler-managed redundant multi-threading for GPUs developed at AMD. The prototype compiler takes an OpenCL kernel as input and automatically transforms the kernel to into one that performs redundant execution within a well-defined protection domain. A performance and power evaluation of this compiler on an AMD Radeon™ GPU is given in the paper by Wadden et al. [20].

Third, metrics, tools, and APIs are required to ascertain what reliability operating point the system should be running at in a given point in time and communicating that information to the flexible RAS mechanism(s) in the system [21]. High-level vulnerability analysis techniques and fault-injection techniques can be useful in the design of such high-level resilience analysis tools [16, 22–24].

As the previous bullets make clear, significant research challenges must be solved in order to realize true resilience proportionality in a system. However, we believe that researchers should use resilience proportionality as a guide for future research to enable breakthrough reliability advances in future generations of microprocessors.

6 Conclusions

This chapter provided a glimpse into the factors and costs involved in providing reliability from the viewpoint of the microprocessor industry. Reliability is an important consideration in the design of processors, especially those targeted at servers and warehouse scale computers. This article makes the case of resilience proportionality as a means to providing reliability in a cost-effective manner and identified areas of research to achieve this goal.

References

1. K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. S. Williams and K. Yelick, Exascale computing study: technology challenges in achieving exascale systems, peter kogge, editor & study lead, 2008
2. A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic Concepts and Taxonomy of Dependable and Secure Computing, in *IEEE Transactions on Dependable and Secure Computing*, (Jan–Mar 2004), pp. 11–33
3. S. S. Mukherjee, J. Emer, S. Reinhardt, The Soft Error Problem: An Architectural Perspective, in *International Symposium on High-Performance Computer Architecture*, 2005
4. R. Baumann, Radiation-Induced Soft Errors In Advanced Semiconductor Technologies, in *IEEE Transactions on Device and Materials Reliability*, 2005
5. C. Constantinescu, Trends and challenges in vlsi circuit reliability, in *IEEE Micro*, (Jul–Aug 2003), pp. 14–19
6. V. Sridharan, N. DeBardeleben, S. Blanchard, K. B. Ferreira, J. Stearley, J. Shalf, S. Gurumurthi, Memory Errors in Modern Systems: The Good, The Bad, and The Ugly, in *International Conference on Architectural Support for Programming Languages and Operating Systems*, 2015
7. S. Mittal, J. S. Vetter, A survey of Software Techniques for Using Non-Volatime Memories for Storage and main Memory Systems, in *IEEE Transactions on Parallel and Distributed Systems*, 2015
8. T. Siddiqua, S. Gurumurthi, A Multi-Level Approach to Reduce The Impact of Nbti on Processor Functional Units, in *Great lakes symposium on VLSI*, 2010
9. M. R. Shaneyfelt, P. E. Dodd, B. L. Draper, R. S. Flores, Challenges in Hardening Technologies Using Shallow-Trench Isolation, in *IEEE Transactions on Nuclear Science,* pp. 2584–2592, 1998
10. R. W. Hamming, Error Detecting and Correcting Codes, in *Bell System Technical Journal*, 1950
11. D. Bernick, B. Bruckert, P. D. Vigna, D. Garcia, R. Jardine, J. Klecka, J. Smullen, Nonstop Advanced Architecture, in *International Conference on Dependable Systems and Networks*, 2005
12. L. A. Barroso, J. Clidaras, U. Holzle, The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, 2nd edn. (2013)
13. D. Lyons, Sun Screen, *Forbes,* 13 Nov 2000
14. A. Biswas, C. Recchia, S. S. Mukherjee, V. Ambrose, L. Chan, A. Jaleel, A. Papathanasiou, M. Plaster, N. Seifert, Explaining Cache SER Anomaly Using DUE AVF Measurement, 2010
15. L. Szafaryn, B. H. Meyer, K. Skadron, Evaluating Overheads of Multibit Soft-Error Protection in the Processor Core, in *IEEE Micro,* pp. 56–65, 2013
16. S. S. Mukherjee, C. Weaver, J. Emer, S. Reinhardt, T. Austin, A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor, in *International Symposium on Microarchitecture*, 2003
17. L. Barroso, U. Holzle, The Case for Energy-Proportional Computing, in *IEEE Computer,* pp. 33–37, 2007
18. Khronos Group, OpenCL, [Online]. Available: www.khronos.org/opencl
19. Heterogeneous System Architecture Foundation, [Online]. Available: http://www.hsafoundation.com
20. J. Wadden, A. Lyashevsky, S. Gurumurthi, V. Sridharan, K. Skadron, Real-World Design and Evaluation of Compiler-Managed GPU Redundant Multi-Threading, in *International Symposium on Computer Architecture*, 2014
21. S. Li, V. Sridharan, S. Gurumurthi, S. Yalamanchili, Software-based Dynamic Reliability Management for GPU Applications, in *Workshop on Silicon Errors in Logic—System Effects*, 2015

22. V. Sridharan, D. R. Kaeli, Eliminating Microarchitectural Dependency from Architectural Vulnerability, in *International Symposium on High-Performance Computer Architecture*, 2009
23. B. Fang, K. Pattabiraman, M. Ripeanu, S. Gurumurthi, GPU-Qin: A Methodology for Evaluating the Error Resilience of GPGPU Applications, in *International Symposium on Performance Analysis of Systems and Software*, 2014
24. S. Hari, T. Tsai, M. Stephenson, S. Keckler, J. Emer, SASSIFI: Evaluating Resilience of GPU Applications, in *IEEE Workshop on Silicon Errors in Logic—System Effects*, 2015

Roadmap for On-Board Processing and Data Handling Systems in Space

Gianluca Furano and Alessandra Menicucci

Abstract The domain of space avionic systems is changing extremely rapidly, compared to other technical domains in space-faring industry, under the pressure of an intense competition, the continuous emergence of new markets and players, the need for cost reduction, as well as an increased obsolescence rate of components and processes. This rapidly changing landscape is as well opening a large amount of opportunities for the space avionic systems: the new high-performance processors architectures and silicon processes, which offer the possibility to integrate different functions until now implemented on several boards either in a single chip (SoC), or in application-specific standard products (ASSP) or in new large FPGAs are allowing multi-fold gains in performances and miniaturization for electronic systems. Another example are digital sensor buses, already heavily used in automotive and embedded applications and now also introduced in space systems to tackle mass, power reduction, increase of accuracy and increase of testability. Reliability and availability constraints remain the main driving requirements for established space hardware manufacturers. Most of the connected world infrastructure, as well as critical services in commercial and governmental domains are still highly dependent from space assets, and avionics-related failures may account for a large part of the system's downtime. In this context, the emergence of space systems based on only Commercial-Off-The-Shelf (COTS) (like Cubesats) is not necessarily helping, since in order to cut costs the rigorous test and quality assurance processes applied to bigger satellite are waived underestimating how unforgiving space environment can be for electronics.

G. Furano (✉)
European Space Technology Centre—ESTEC—European Space Agency,
Keplerlaan 1, 2201AZ Noordwjik, The Netherlands
e-mail: gianluca.furano@esa.int

A. Menicucci
Faculteit Luchtvaart En Ruimtevaarttechniek, TUDelft, Postbus 5058,
2600GB Delft, The Netherlands
e-mail: a.menicucci@tudelft.nl

M. Ottavi et al. (eds.), *Dependable Multicore Architectures at Nanoscale*,
DOI 10.1007/978-3-319-54422-9_10

1 Technology State of the Art and Challenges in Europe

Data Handling Systems and On-board Computers encompass a vast range of functional blocks including On-Board computers, Telecommand and Telemetry Modules, Data Storage and Mass memories, Remote Terminal Units, Communication protocols and Buses. With some different denominations and arrangements, these elements are common to all space projects and are subject to a demanding set of evolving requirements from different class of missions (like Science, Exploration, Earth Observation and Telecommunications) and intrinsically linked to software technology, including validation techniques.

In addition to the steady evolution of the "classical" requirements, requiring an increasing processing power, reduced mass, volume and power budgets, new driving requirements are identified during the definition phase of the next generation programmes. Such requirements are not only related to the implementation of functional services linked to on-board communication standards but also to architectural and development paradigms for the system and application software.

A new commonality exhibited by Telecom and Earth Observation missions is the need to produce multiple satellites or platforms in a restricted time scale and this implies the application of industrial procedures that are increasingly implemented in space programmes (e.g. constellations). Science missions require enhanced modularity and multi-instruments support whereas Earth Observation missions need very high data throughput links and increased on-board storage memory capacity. Furthermore, Earth Observation missions share with Telecom projects architectures that implement security solutions to protect and secure the Payload (P/L) data and the Spacecraft (S/C) control and monitoring functions.

Since space is a highly regulated market, with many interactions with government and export licenses, underpinning the above is the need to provide European-based solutions at competitive costs and this aspect is being used to drive the research and development approach. In particular, emphasis is being placed on the development of standard building blocks which may be used across multiple missions and for defence systems (for which specifics may exist). Such an approach requires a clear strategy in terms of agreed avionic architectures, functional elements and the standardization of interfaces and protocols. Cooperation is also required with software disciplines to ensure that hardware elements can be integrated with a minimum of effort to be reinforced through coordination with the software roadmap.

The Data systems and On-Board Computers technologies, with particular focus at European landscape, are reviewed in this article on the basis of the present status, current developments and future mission requirements. The technologies for the current and future missions are mapped, key technology development areas are identified and a broader vision of the long-term technology trends is developed.

Space industry and Agencies have been recognizing already for quite some time the need to raise the level of standardization in the spacecraft avionics systems in order to increase efficiency and reduce development cost and schedule. This also includes the aspect of increasing competition in global space business, which is a challenge that European space companies are facing at all stages of involvement in the international markets. A number of initiatives towards this vision have been initiated by the European industry and European Space Agency' (ESA) R&D programmes. The most successful one is the Space Avionics Open Interface Architecture (SAVOIR) initiative that sees the participation of ESA and several European companies and national agencies.

Nevertheless, lack of new programmes, delayed funding for others, as well as a general lack of capability from programme managers and their political stakeholders to accept and foster innovation has created a demand in space circles to extend the life (both as in-space lifetime and as commercial product useful life) of avionic platforms and look for ways to reduce funding in less critical systems by pursuing more commercial parts and manufacturing processes. Budget constraints put in place over the last few years are finally affecting the basic electronics R&D and thus all new designs for satellites and other spacecraft at the component level. The long life cycles of space products typically lag behind terrestrial systems by about three to ten years mainly due to the rigorous qualification requirements for electronics in the high radiation environments of space. The uncertainty about what will be funded in the long run is also making it difficult for suppliers to plan future development strategies.

While some programmes such as deep space missions (like future ESA's Jovian platform JUICE) and manned spacecraft platforms will still require the best class of radiation hardening of their electronic components, there is an increased demand to apply solutions requiring less protection in less critical applications. This puts pressure on designers and the dwindling number of rad-hard EEE part suppliers to meet these cost reduction demands while maintaining the reliability of their parts for space missions. Conflicting with these demands are requirements to guarantee a longer operational lifetime for the satellite. An example of rad-hard budget constraints for satellites is the request of increased life span out of existing designs because funding for R&D for new programmes is delayed or reduced. In the past, Telecom platforms were typically required to last for 15 years, but now programme managers want to extend them as long as 18 years, with long phases of Electrical Orbit Raising that increase component's dose up to 25%. The pressure is then on the component, units and system suppliers to guarantee technology can reach those expectations, particularly for systems looking for an extension of legacy designs.

A communications satellite being sent up for 18 years still needs full space-qualified parts, but for shorter life platforms, even NASA is looking to do more with less. NASA Jet Propulsion Laboratory (JPL) has been flying 30 Krad Commercial-Off-The-Shelf (COTS) boards in satellite experimental platforms for years, while NASA NEPP, as a follow-on to an internal NASA EEE parts workshop held in 2013, is hosting several open workshops entitled "EEE Parts for Small Missions". Small Missions are loosely defined as those under 500 kg, but the emphasis here is on under 100 kg. This is precisely the market sector that is now booming thanks to the announced telecom "Mega-constellations" [1] from OneWeb and others. The key here is tailoring EEE parts approaches based on mission risk and expectations. This includes "traditional" (science) and "non-traditional" (demonstration) missions with CubeSat/Nanosat/Microsat electronics as prime research area (but still far from any useful operational target).

A specific case of rapid (for space electronic standards) introduction of 'new' components in space avionics is use of FLASH memories [2]. NAND flash is currently the most suitable solution for nonvolatile storage in embedded applications and it is gaining access to in safety-critical applications, thanks to their high storage density, low power, low cost, and high data throughput. However, NAND flash research and literature in the safety-critical environment is not as established as in the commercial applications. As a matter of fact, for the specific case of space applications, NAND flash is struggling to keep pace with those advances for multiple reasons.

2 On-Board Computers and Data Systems Architectures and Their Generic Specifications

Since many years space industry and Agencies have recognized the increasing need to raise the level of standardization in spacecraft avionics systems in order to increase efficiency, reduce development cost and schedule, and operate in more optimized development and verification environment. The objective of reuse, whereby standardized building blocks may be developed once and used across multiple missions is also high on the agenda. Such an approach requires the rationalization of architectures such that recurring elements may be identified and functionally specified along with the interfaces and protocols for interconnection.

To provide a reference for all the avionic standardization efforts, ESA SAVOIR initiative [3] has defined a generic functional architecture for an On-Board Computer (OBC) and Platform Data System that is here below shown where all the functional blocks of a Platform Data System are shown together with the typical redundancy philosophy.

The mission domains considered by the SAVOIR standardization initiative are as follows:

- Science and Earth Observation missions,
- Telecom missions and
- Commercial Earth Observation missions.

A generic functional specification for the OBC has been produced and reviewed in a public consultation in spring 2015. Generic specifications for RTU and Solid State Mass Memories are under preparation. A real system may include more functionality than what is covered in the SAVOIR generic OBC functional specification (e.g. some OBCs may for instance combine the platform and payload data storage in which case the functional block "Payload Data Storage" of Fig. 1 (*right* side—*centre*) is included in the OBC). Alternatively, for mission scenarios different

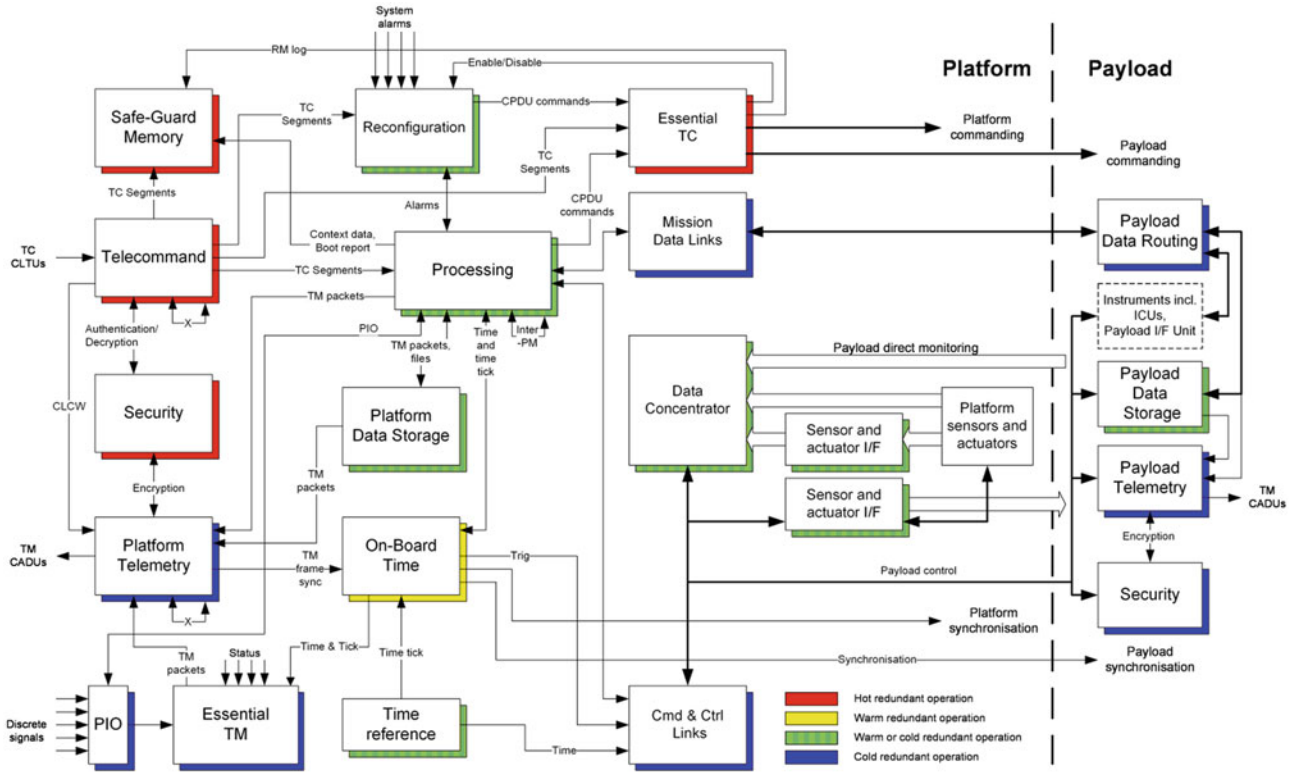


Fig. 1 SAVOIR functional reference architecture

from the ones considered by SAVOIR (that are Earth Observation, Science and Telecommunications) not all the building blocks or functions depicted in Fig. 1 are necessary or, if present, a different redundancy scheme with respect to the one shown can be adopted.

The following spacecraft platform functions are presented in Fig. 1:

- Telecommand reception, decoding and distribution.
- Security function that protects the spacecraft from receiving unauthorized commands and that provides optional decryption and encryption of data sent on the TM/TC link.
- Telemetry Transfer Frame generation and coding.
- Essential TM function, collecting essential data and generating data packets for the TM Encoder.
- Essential TC function, distributing pulse commands to control vital spacecraft functions.
- Parallel I/O to support the acquisition of discrete essential spacecraft data.
- On-Board Time management, providing a time counter and generating synchronization events.
- Platform Data Storage function for storage of data needed for the spacecraft operation.
- Safeguard Memory function for storage of vital spacecraft data that is needed by the processing function.
- Reconfiguration function that maintains the operation of the processing function even in case of errors.
- Processing capability to store and execute Execution Platform and Application software.
- Communication, separated into Mission Data and Cmd & Ctrl communication systems, allowing the processing function to communicate with platform sensors and actuators and with the spacecraft payload.
- Data Concentrator function for handling the monitoring of spacecraft sensors (usually implemented in one or more RTU/RIU).
- Sensor and Actuator Interfaces for interfacing the physical sensors and actuators.
- Payload data routing function for routing monitoring and control communication to and from payload units.
- Payload Data Storage function, for storage of payload TM data during periods of no ground station contact. Optional function.

Figure 1 also shows the redundancy concept for all the functions. To avoid single-point failures of the system all functions are required to be internally redundant. Redundancy can then be operated in three different ways as follows:

Cold redundancy

In cold redundancy, there is one Active part of a function that is operating and its redundant part which is not operating. This means that the redundant part can be powered or unpowered, the latter case being also physical cold redundancy. An example is the Telemetry function where there is only one TM Encoder that generates data to the RF system, but where the other TM Encoder does nothing but may be powered or unpowered depending on the selected physical implementation.

Warm redundancy

In warm redundancy, there is one Active part of a function that is operating and its redundant part which is operating but with reduced functionality, possibly as a back-up that is ready to take over in case the Active part fails. An example can be a back-up mass memory data storage that is used to store a copy of critical TM data but as long as the Active mass memory data storage operates correctly, data is never downloaded from the back-up memory.

Hot redundancy

In hot redundancy, there are two or more simultaneously active parts that operate in parallel and that can even produce parallel outputs. An example is the TC Decoders which operate in parallel on the received data. If they both have the same VC ID, they could actually also output data in parallel.

Warm-standby redundant systems are characterized by the fact that the spare module is active in parallel with the main module, and in the ideal case, receives and processes all data of the connected peripherals. Built-in test and automated self-monitoring mechanisms enable self-governing fault detection. This results in autonomous switch-over if the active master fails, via, for example, an external trigger or watchdog mechanism. The switch-over time (and thus the relative MTBI) compared to a similar cold-standby system is significantly reduced in this process.

For some applications (human space vehicles or landers), architectures based on triple or even quad redundancy with majority voting are proposed and implemented. Fast inter-communication buses and voting circuitry are usually used to cope with the required reduced latency of such architectures especially in critical phases as landing or docking. For these applications, the term high-available computer is used instead of high reliable computer to indicate that in certain phases of the mission outages of any duration in time (e.g. reconfiguration from main to redundant stream) are not accepted. To be underlined that these types of applications are currently excluded from the SAVOIR perimeter; the possibility to include them in future revision of the specification are under discussion.

3 A Historical Perspective on Space Microprocessors

The most remarkable example of long-term R&D process for complex EEE parts in Europe is the development of radiation hardened by design SPARC processors for space.

In late 1990, the European Space Agency started the ERC32 project to set the pace for the development of higher performance processors for ESA and European avionics. At the time the 32-bit Microprocessor and Computer Development programme started the availability of ESA-supported processors was limited to 16-bit processors.

ERC32 was a major success for European manufacturers (French electronic company TEMIC, later absorbed by ATMEL), with more than 15,000 flight units sold and is still the main workhorse processor in many space missions (one for all the European Ariane-5 launcher), but by mid 1990s it was clear that a successor project was necessary to keep the innovation pace of avionic systems, thus in 1997 ESA started the development of the LEON.[1] The choice of open instruction set architecture (ISA) was immediately seen as first-level requirement for this development (both to avoid availability issues and to allow customization), but suitability for use in space was also a major consideration.

For its processors since LEON ESA has chosen to use SPARC Version 8.[2]

SPARC had several recognizable advantages as follows:

- At the time of this choice, it was perhaps the only fully open ISA with significant backing.
- It was a reasonably simple ISA (RISC), friendly to lower effort implementation.
- Workstations existed using SPARC v8 compatible processors, so cross compiling could be avoided (at that time virtualization/real time emulation were still not possible).
- Development tools for SPARC were reasonably mature, and a gcc compiler was existing and well maintained, although all the tools were presumably more oriented towards server/workstation workloads.

However, SPARC also had some disadvantages, whose relative weight was very difficult to evaluate back in 1997 as follows:

- By 1997 (with the introduction of the Pentium II) the future of SPARC in workstations would have begun to be under question (making the cross compiling issue somewhat questionable).
- SPARC was not being broadly adopted as an open ISA, so there was not a significant commodity effect (this was probably not a major consideration for LEON, but influences the availability of software tools. The openness of SPARC also has limited advantage in terms of patents).
- As a classic RISC SPARC had somewhat poor code density (code size used to be a significant factor for space-based computers, since it is linked to memory size).
- SPARC was not being broadly adopted for use in embedded systems. (This would have influenced the availability of development tools suitable for such systems).

[1] Nicknamed after Luc Besson's famous killer.

[2] 1991/1992 based on the copyright notice in The SPARC Architecture Manual: Version 8, the base ISA was released in 1987.

- SPARC's register windows would have increased minimum core size, slightly increased hardware design complexity and involved more complex development for tightly constrained real time operation.
- SPARC included less useful features like tagged arithmetic. (This was probably not significant since a subset of the architecture could be used).
- SPARC's instruction format was somewhat less regular than other RISCs. (This is a rather trivial objection; the size/power difference between instruction decoders for an Alpha-like ISA and SPARC would be less than 1%.)

With somewhat guaranteed use by the ESA of whatever architecture was chosen and the special requirements for space-based systems, it might seem that a custom, fully space-optimized ISA might have been practical at the time.

But clearly, a custom, open ISA would have significant disadvantages as follows:

- The lack of existing development tools would have added cost and linked compiler development to hardware development. This at the time was considered the strongest argument against a custom ISA. The switch towards a higher ladder of coding abstraction (with rapid introduction of C language coding) had already shown a trend towards the increasing of software complexity, schedule risks and associated development costs. Non-open tools for SPARC development would have represented some risk to LEON, but this could rightly have been considered a minor issue.
- The lack of same-ISA workstations would have added complexity and cost to initial development.
- Even excluding the cost of producing software development tools, developing a new ISA has significant cost and risk and getting consensus without design-by-committee effects is challenging and second system dangers may have been significant.
- Any increase in (early) cost and risks may have disproportionately increased the probability of project failure. On the other hand, a custom ISA could have been more fit for the purpose and might have been able to generate more external adoption in aerospace both of the ISA and of chips developed for the ESA which would have increased the quality of software, increased the testing of implementations and reduced hardware costs.

Clearly, if you choose a custom ISA, you have to create everything yourself, starting from the entire chip design (instead of being able to use existing building blocks as with SPARC), compiler, all software including the OS, cross-compilation and other equipment you need to develop software for this new ISA. You also lose the convenience of having off-the-shelf hardware for the thousands of systems you need on earth, for development, testing, etc. And you lose the thousands of man-years already invested in testing the SPARC architecture and software. By 1991, any bugs in SPARC were well known; SPARC software was mature and in use everywhere. A custom ISA might gain a bit more efficiency, but you increase the difficulty of the project by several orders of magnitude.

So how did the ESA come to choose SPARC over a custom ISA (or perhaps negotiating a limited perpetual license for another RISC ISA)?

At the time ESA performed two architectural studies, evaluating processors such as MIPS, THOR, MC68020, I386, NS32. ESA also invited industry for several round-table discussions. Finally, the agreed selection criteria that finally led to SPARC choice were as follows:

1. Open architecture without patents or license fees
2. Well designed and documented
3. Easy to implement
4. Established software standard
5. Available design.

If those were selection criteria, it can be easily seen that a custom ISA would not have satisfied some of them. The final report for the ERC32 programme states that:

> Furthermore, it was requested to "reuse" an existing processor architecture in order to minimize both software and hardware development cost. Performed ESA and industrial studies resulted, at that time, in the selection of SPARC instruction set architecture as the baseline. This was in order to e.g. simplify bread-boarding and software development.

So, ESA was adamant in deciding to specifically require an existing ISA to be used.

Designing SPARC processors can be done without any licenses whatsoever. This is indeed why Jiri Gaisler, the mind at ESA behind this whole selection process,[3] has selected SPARC for the development of LEON. If we consider how many times in the later years companies like Intel, MIPS and ARM have sued smaller companies that developed processors using their architecture this was indeed a wise choice.

For example, in February 2002 the legal battles between Lexra/MIPS and picoTurbo/ARM have both ended with a complete defeat of the two CPU-cloning companies (Lexra [5] and picoTurbo). Both companies, after long and expensive lawsuits, have been shutdown and their customers transferred to MIPS/ARM.

Commenting those events Jiri Gaisler said, "*More than ever, I'm happy with the decision to go SPARC. And many thanks to Sun and SPARC International for the open license*!"

Nevertheless one of the major step forward in LEON development was the fact that it overcame what were the recognized ERC32 limitations for future upgrades:

[3] And later founder of Gaisler Research [4], that made LEON architecture and ancillary IP cores available to worldwide space industry.

- Proprietary design at schematic/layout level
- Difficult to port
- Complex interface
- 20 MHz limit due to memory interface.

LEON project goals were established to provide headroom not only in performances, connectivity and reliability with respect to ERC32, but also to suit the industrial model of European space electronic manufacturers.

- European design
- Radiation hard and SEU free
- Standard interfaces
- Modular
- Portable
- Written in VHDL
- 100 MIPS, 20 MFLOPS.

These latter characteristics, and not the ‘brute’ performances were the key of LEON’s commercial success and widespread (for space avionic terms) adoption.

LEON2 was designed for a single function (processor), targeting an Application-Specific Standard Product (ASSP) that came in 2009 with Atmel’s AT697F, but it has been increasingly being used as SoC platform since the very beginning. This generated a positive feedback with availability of IP libraries and use of commercial ones in several space grade SoCs. The availability of large rad-hard PLDs (especially with introduction of ACTEL/Microsemi RTAX series) made the rest, since now space components and units manufacturers have a very efficient SoC platform that helps in minimizing design and SW development work, resulting in many first-silicon-good ASICs.

The LEON ‘ecosystem’ is now grown up (mostly thanks to ESA and COBHAM-Gaisler support) and thanks to its portability and CAD tool independence allows to support of both ASIC and FPGA technologies, taking the best of both worlds.

LEON cores performances on high-end modern FPGAs allow its use as soft core in payload processing (several examples from VENUS-express onward), its set of coherent IP interfaces allows build of template designs for common FPGA boards and bespoke SoC with a manageable effort (Thales-Italy EPICA-Next was a very successful example, powering computers in Iridium-Next constellation) and the uniform method for HW & SW debug allows to maintain coherency between the different instantiations, maintaining portability of SW and know-how across industry. Last generations LEON rich functionality (like MMU and MP support) are allowing also daring avionic solutions like the star tracker processing in OBC or time and space partitioning (Table 1).

Table 1 ESA and European space processors development timeline

Year	Event
1989	MDC281 (1750) clone, 2.5 um CMOS/SOS, 0.5 MIPS
1991	MA31750, 1.5 um CMOS/SOS, 2 MIPS
1992	SPARC architecture selected as ESA baseline in competition with MIPS, NS32, M88 K, AMD29 K
1995	3-chip ERC32 (SPARC V7), 10 MIPS, 0.8 um—ISS control computer, 10 missions
1998	single-chip ERC32 (TSC695), 15 MIPS, 0.6 um—for more than one decade standard processor for all ESA missions, more than 15,000 chips sold! Used by NASA, China, India, Israel
2000	First LEON1-FT, 0.35 um, 50 MIPS, 0.5 W (commercial ASIC)
2002	First LEON2-FT, 0.18 um, 100 MIPS, 0.6 W (commercial ASIC)
2004	First LEON3-FT, 0.20 um, 150 MIPS, 0.4 W (commercial ASIC)
2007	LEON3FT quad-core, 90 nm, 4 × 500 MIPS, 3 W (commercial ASIC)
2008	First flight-worthy LEON-2FT SoC 180 nm rad-hard process (COLE by RUAG-SE)
2010	First LEON4 (LEON4-DEMO/GR-LEON4-ITX)
2012	First flight-worthy LEON-3FT Dual-Core SoC (GR712 by Cobham-Gaisler)
2012	First quad-core LEON4FT, 45 nm commercial process (LEON4-N2X)
2016	LEON4FT quad-core EM, 65 nm rad-hard process, quad-processor LEON4FT SoC running at 250 MHz, with 425 DMIPS/core

4 Emerging Trends: Microcontrollers

In the current space market the increasing systems complexity poses great challenges to architects, designers and verification engineers. Systems requirements are continuously asking more functionality with less power, less resources, less time and cost, forcing suppliers to shrink their schedules. One important aspect that is becoming more and more critical nowadays is a ‘first time right’ design that can be scalable and adaptable to special needs, leveraging on the reuse of building blocks which are highly configurable and already validated.

One of these building blocks that may make a real change in the current space market is a microcontroller (MCU). In the commercial market the microcontroller has proven its value filling up nearly every electronic device from home appliances to hand gadgets, and now booming thanks to Internet-of-Things devices. Quoting from Wikipedia[4]: *in 2002, about 55% of all CPUs sold in the world were 8-bit microcontrollers and microprocessors. Over two billion 8-bit microcontrollers were sold in 1997, and According to Semico, over four billion 8-bit microcontrollers were sold in 2006. More recently, Semico has claimed the MCU market grew 36.5% in 2010 and 12% in 2011.*

[4] http://en.wikipedia.org/wiki/Microcontroller.

The great popularity in the commercial market is essentially based on their flexibility, the low cost and the great variety of technical solutions these devices may offer, with many of them readily available as 'reference designs' that include hardware and software examples. It is also important to notice that a plethora of tools for developing MCU-based solutions are widely accessible to a huge user base (think about AVR-based Arduino's success), accelerating development cycles and bringing down costs and time.

The space market has, so far, been confined to either use of a full fledged microprocessor, like the LEON discussed above, or handcraft increasingly complex state machines in overly loaded FPGAs [6]. There is no man in the middle with reasonable radiation hardness available to system designers to offload the complexity of current designs and reuse is limited to functions or IP cores that are often not posing the most complex design or verification challenges (memory controllers, PWMs, serial communication ports, …). A microcontroller may come to the rescue though (and a recent ATMEL announcement has raised lots of hopes in designers [7]), providing a solution that fits in between the overly loaded FPGAs and the hugely complex microprocessor, providing another layer of abstraction to tackle complexity. Moreover, a microcontroller-based solution may quickly respond to rapidly changing needs and securing development and verification efforts, delegating repetitive and highly parallelizable functions to the FPGAs (packet switching, low level protocols, hardware interface, interconnection,…), while keeping the algorithmic intelligence of decision-making. Embarking on building a space-rated microcontroller is indeed a huge effort and is by no means an easy task, therefore, we are currently proposing to change perspective and look towards a potentially more viable solution, more portable and flexible than a microcontroller: a soft core.

Quoting again Wikipedia[5]: a soft microprocessor (also called softcore microprocessor or a soft processor) is a microprocessor core that can be wholly implemented using logic synthesis. It can be implemented via different semiconductor devices containing programmable logic (e.g., ASIC, FPGA, CPLD), including both high end and commodity variations. With the new available technologies, where available gates per device is increasing and on-board memory is abundant, thanks to emergence of DRAM for space use, a soft core, with a small footprint, may be a key element for system designers to reduce system complexity and increase reuse.

5 Use of Programmable Logic Devices in Hi-Rel Space Avionics

Microsemis range of Radiation Tolerant FPGAs are being more and more adopted for the full range of space applications. Also increasingly adopted is their first flash-based FPGAs for radiation applications, RT ProASIC3. RT ProASIC3 now

[5] http://en.wikipedia.org/wiki/Soft_microprocessor.

has flight heritage on several science missions, including the International Space Station and at least two NASA missions: IRIS (LEO) and LADEE (Lunar orbit). ESA adopts it for Insight Mars surface payload, as well as two Exomars 2018 systems. RT ProASIC3 also has flight heritage on several international LEO remote sensing missions, and will be deployed in commercial communications missions in the near future. It is also expected a relatively easy adoption of RTG4, Microsemis next generation FPGA family for radiation environments which uses a 65 nm low-power flash process, and is immune to changes in configuration due to radiation effects and goes well beyond ProASIC3 in terms of TID resilience.

In recent years Europe has taken a stronger position related to removing all ITAR products from space systems, and this has led to directives to eliminate U.S. ITAR products even if it means reducing performance and quality. Finally, due to the critical situation with Russia, one of largest growing space market segments is no longer available due to export restrictions. The Russian space market has been very quick to react and is currently looking for non-U.S. alternatives, similar to what Europe is doing. Increasing partnership with Russia (Exomars, Lunar Lander) and China (Chang'e-3, SMILE) will in any case put pressure towards EAR/ITAR free designs, also to boost EU industries competitiveness.

Nevertheless ITAR is subject to the U.S. Department of Commerce's Export Administration Regulations (EAR) and classified under ECCN 9A515.x or similar since November 2014. Most of the Rad-Hard FPGAs along with associated SW, support and documentation requires then a license before manufacturers can export them.

The U.S. is currently authorizing exports to China/Russia on a case by case basis. Generally speaking, for example, exports for native Chinese programmes are not allowed. However, export licenses for international programmes where China is one of several participants may be approved. The definition of what is an "international programme" has yet to be fully tested.

The EAR, unlike the ITAR, also includes a concept called "*De Minimis*". The ITAR controlled parts is effect from the time they leave the US to the time they are launched into Space. On the other hand, the EAR's reach eventually ends. EAR Part 734.4(d) provides that re-exports of a foreign made commodity incorporating controlled U.S. origin commodities valued at 25% or less of the total value of the foreign made commodity are not subject to the EAR. Note, however, that 734.4(a) provides that this rule does not apply (and the re-export is subject to the EAR) when destined for a country listed in EAR Country Group D:5 of Supplement No. 1 to part 740 of the EAR (as China).

The EAR also allows for something else the ITAR does not. The ITAR required US companies to have a Purchase Order in place before applying for an export license. There is no such restriction for EAR license. So EU companies could easily apply for a license in advance of a purchase order and see how this ends up.

5.1 Performance Critical Functions. A Use Case for Future Avionics

Due to recent technological developments, high-performance floating-point signal processing can be easily achieved using FPGAs, and for the first time this holds also in space grade hi-rel applications. To date, virtually all signal processing has been implemented using ad hoc fixed-point operations.

Numerous studies have evaluated application performance for FPGAs and GPUs. Much work has been focused specifically on image and video processing. One challenge that makes such exploration difficult is that there is rarely a globally optimal device for a particular application. Instead, applications generally have a set of Pareto-optimal implementations that trade-off numerous metrics such as performance, power, energy, cost, size, reconfigurability, application-design complexity, fault tolerance, etc. Furthermore, such exploration is complicated by numerous use cases.

In space applications, autonomous navigation and localization are based on the extensive use of computer vision with the following constraints as follows:

- Need to minimize resources (power consumption, mass and volume).
- Space-qualified components provide limited CPU/memory performances.
- Single-chip power/clock energy limited by the available solar power and power dissipation capability.

As a result, for example Martian Rovers with autonomous guidance and obstacle avoidance spend more than 50% traverse time in data processing while not moving. Foreseen solution is to improve computer vision capabilities by means of custom-designed vectorial processing (FPGAs). In descent and landing applications the frequency and/or performances are often degraded or traded during the synthesis of the Image Processing algorithms to fit in space-qualified FPGA resources.

A possible architecture of a vectorial processing for image pattern recognition/autonomous navigation is depicted in Fig. 2.

A full trade-off of the Image Processing algorithms with respect to potential for realization in space-rated (or space-capable) devices is necessary and together with the implementation of the algorithms in a demonstration setup will set a clear path for this type of applications. These type of application will be the ones setting the need for future high-performance processors and pushing the trade-off between single-core and homogeneous multi-cores.

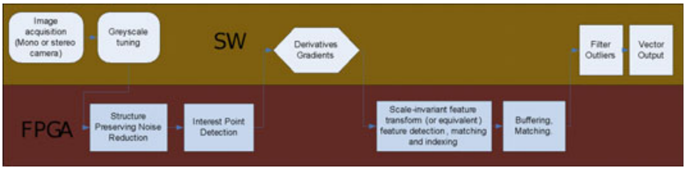


Image acquisition (Mono or stereo camera)
Greyscale tuning
SW
Derivatives Gradients
Filter Outliers
Vector Output
FPGA
Structure Preserving Noise Reduction
Interest Point Detection
Scale-invariant feature transform (or equivalent) feature detection , matching and indexing
Buffering, Matching

Fig. 2 Possible HW/SW distribution for an image recognition algorithm

5.2 Future Trends in Processors

In the future a move from compatibility based on ISA to higher level compatibility based on an API. This type of approach has been already used in launchers, to maintain compatibility with validated software routines. Better design efficiency and long-term compatibility can be achieved using heterogeneous cores. The cellphone's use of many specialty cores working in concert may be a good model for future multi-core designs.

In the past, often performance of the available space grade building blocks (like processors, PLDs) was compared against Desktop-computing CPUs. There was plenty of quoting of Moore's Law. The last five years, with the explosion of the mobile electronics market and the approach of the IoT (Internet of Things) applications have brought the focus on embedded applications and their trends. It is becoming harder and more expensive to scale chips, and Moore's Law, the engine that drives computing and electronics, seems to be winding down. But the demand for faster, more capable processors that use less power remains driven by everything from luxury sedans to smart watches.

Embedded processors do not have the same name recognition as the Intel Core processor, but there are many more of them toiling away in everyday devices such as ATMs, networking and communications gear, cars and other vehicles, and of course the Internet of Things. Of the total processor IP market (15.3 billon chips shipped in 2014), mobile is the biggest at 6 billion, but embedded is nearly as large and growing faster, followed by enterprise (including PCs), consumer and flash storage. As these applications grow more complex (and demanding), the industry is shifting from general-purpose processors to highly application-specific processors packing more functions onto a system-on-chip (SoC).

Until recently chip-makers could rely on physical scaling to meet most of these challenges. By jumping to the next node, they got better performance, lower power and more transistors to work within a given area to add new features. But Moore's Law is running out of steam and the cost per transistor increased at 20 nm, and again at 14 nm, because of additional lithography steps. It used to be that everybody moved to the next node because it was cheaper, better and faster, and that was the only extra technology step needed, but now things are getting much more complicated.

These complex SoCs also require knowledge of the intended application, access to lots of intellectual property and a complete platform including software. Many companies are also choosing to design their own custom cores, in particular on 64-bit ARMv8 because the ecosystem is very familiar to the engineers and programmers that are designing these systems. Examples include AppliedMicro's X-Gene, Cavium's ThunderX, Broadcom's "Vulcan" core and a Marvell custom ARMv8 core. ARM SoCs are increasingly using APIs to offload tasks from the CPU to the GPU, DSP (Digital Signal Processor), or specialized image or vision processing engines that can handle them more efficiently. In some cases, they are

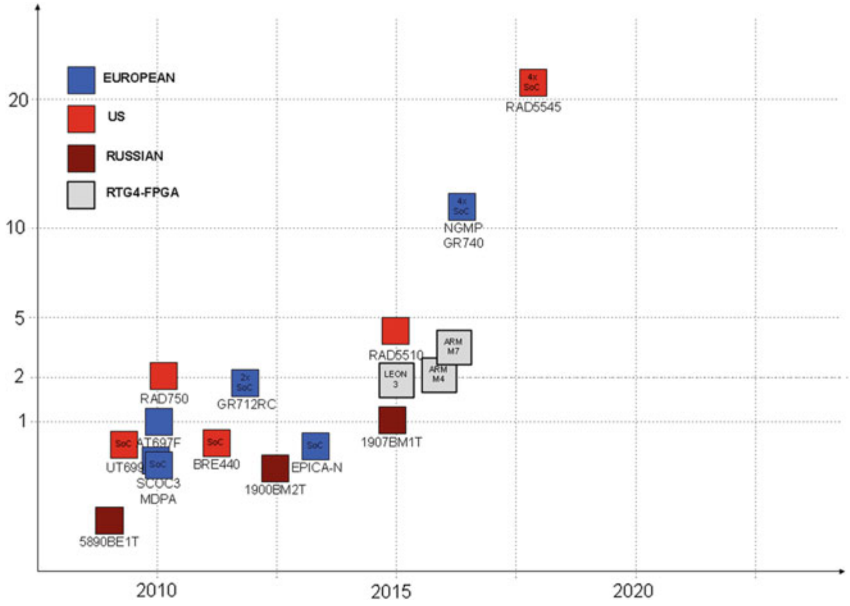


Fig. 3 Present and future processors for space

using network-on-a-chip (NoC) IP from the likes of Arteris, NetSpeed and Sonics to stitch together all these various blocks.

The quest for new microprocessors in space domain are as follows:

- More functionality in a reduced component count, pushing on the SoC trend;
- More processing power in Payload Controller for more data processing capabilities with low power;
- More processing power in Mass Memory for more data handling functionality (e.g. CFDP) and higher level programming abstraction by using more thorough operating systems;
- More processing power in Instrument Control Units that would allow a full equipment saving if Instrument Control Unit and Data Processing Unit could be merged.

LEON2- and LEON3-based SoC are now the main workhorses for all the major manufacturers of On-Board Computer Systems in Europe.

Figure 3 indicates the evolution and the trend in term of MIPS (Y scale is roughly proportional to DMIPS, with AT697F at 140 DMIPS) of the microprocessors used in European Space products in the current projects.

Airbus E3000 platform is abandoning MA31750 16-bits microprocessor in favour of SCOC3 [8, 9] and the ATMEL TSC695F is gradually going into obsolescence although used in missions currently on orbit or under final system level qualification phase (Lisa Pathfinder, Sentinel(s), SWARM, Earthcare) and in

Launchers (Vega, Ariane). For our current knowledge we can safely foresee that all new central on-board computer architectures will be LEON-based.

AT697F has represented a natural evolution of the ERC32: it provides higher performances with increased functionalities but it is still a classic microprocessor in the sense that memories (volatile and not volatile) and application-specific peripherals have to be added to build a complete Processor Module or Single Board Computer Core. AT697 has been selected for ERNOBOX built by Airbus Space Transportation and it has already flown on the International Space Station and on Proba-2 (as version E). AT697 is the microprocessor for GAIA OBC and is used as core processor for Payload Computers (e.g. SWARM or Sentinel −1 for the Instrument Control Module and as well as in EXOMARS 2018 rover computers from RUAG-SE).

However, we have to recognize that LEON-based architectures were far more successful when integrated in system on a chip solutions where together with the typical functionalities of a microprocessors there are other SMU/OBC/CDMU functionalities as CCSDS TM/TC interface and other external interfaces as SpaceWire, MIL-STD-1553B, CAN, Packet Wire and distribution of synchronization signals.

Also in US the trend is towards highly functional SoC. The RAD5500 is a radiation-hardened 64-bit multi-core processor platform manufactured by BAE Systems Electronics, Intelligence and Support with Power Architecture-based technologies from IBM and Freescale Semiconductor. Successor of the RAD750, the RAD5500 processor platform is for use in high radiation environments experienced on-board satellites and spacecraft. The RAD5500 platform supports VPX high-speed connectors, DDR2/DDR3 memory, serialize/deserialize (SerDes) and SpaceWire IO.

The RAD5500 family of radiation-hardened processors use the QorIQ Power Architecture with processor cores based on versions of the Freescale Technologies e5500 core. The RAD5510, RAD5545 and RADSPEED-HB (Host Bridge) are three system on a chip processors implemented with RAD5500 cores produced with 45 nm SOI technology from the IBM Trusted Foundry.

The RAD5510 processor employs a single RAD5500 core and is intended for medium processing capability in environments that require low-power consumption. This processor provides up to 700 MIPS and 466 MFLOPS of performance. The RAD5545 processor employs four RAD5500 cores, achieving performance characteristics of up to 5200 MIPS and over 3700 MFLOPS.

Based on the RAD5545, the RADSPEED-HB is intended for host processing and data management support for one to four RADSPEED DSPs. The RADSPEED-HB replaces a secondary DDR2/DDR3 memory interface connection found on the RAD5545 with connections for RADSPEED DSPs instead. (Note that RADSPEED DSPs are entirely different processors that are specialized for digital signal processing and are not to be confused with the RADSPEED-HB, which serves as a host bridge).

All European major space prime integrators have developed one (or more) LEON-based SoC:

Fig. 4 OSCAR flight OBC during final integration (courtesy AIRBUS)

- SCOC3—(LEON-3FT + GRFPU developed by Airbus-F on ATMEL ATC18RHA);
- MDPA—(LEON-2FT + DSP developed by Airbus-D on ATMEL ATC18RHA);
- COLE—(LEON2-FT developed by RUAG-S on ATMEL ATC18RHA);
- EPICA-NEXT—(LEON3-FT developed by TAS-I on ATMEL ATC18RHA).

There are also general-purpose SoC from independent vendors that are enjoying a very good market success like GR712RC—(2xLEON3-FT developed by COBHAM on Ramon Chips RadSafeTM library on Tower Semiconductors standard 180 nm CMOS technology).

All these SoC products have already flown in many missions or are planned in major recurrent platforms:

- SCOC3 as main core of the Airbus Computer called OSCAR (Optimized Space Computer Architecture with Reconfigurable LEON3), and on Telecom Eurostar NEO and QUANTUM (SSTL);
- MDPA as P/L controller in Alphasat and as Mass Memory controller;
- COLE as core processor of the SGEO platform SMU;
- EPICA-NEXT in Iridium-Next 88 computers and in all the future Spacebus Neo SMU-V2.2 computers (Fig. 4);
- GR712RC as payload processor in MASCOT, SEOSAT, JUICE and as on-board computer central processor in SAT-AIS (ARTES-21).

The introduction of Microsemi's RTG4 might provoke a landscape change in this market, allowing to have FPGAs used (either as main or as interface coprocessor) in OBC or Payload Computers. Figure 3 shows the possible performance position of RTG4-based SoCs with respect to current and planned LEON-based ASSPs. RTG4 FPGAs integrate Microsemis fourth-generation flash-based FPGA fabric and high-performance interfaces such as serialization/deserialization (SERDES) on a single chip while maintaining the resistance to radiation-induced configuration upsets in the harshest radiation environments.

RTG4 FPGAs are manufactured on a low-power 65 nm process with substantial reliability heritage and will be qualified to MIL-STD-883 Class B, Microsemi will seek also QML Class Q and Class V qualification. RTG4 FPGAs are advertised by Microsemi as immune to radiation (SEU) induced changes in configuration 1, due to the robustness of the flash cells used to connect and configure logic resources and routing tracks.

No background scrubbing or reconfiguration of the FPGA is needed in order to mitigate changes in configuration due to radiation effects. Data errors, due to radiation, are mitigated by hardwired SEU resistant flip-flops in the logic cells and in the mathblocks. Single Error Correct Double Error Detect (SECDED) protection is optional for the embedded SRAM (LSRAM and uSRAM) and the DDR memory controllers.

Given the limitations of available ASIC technology, ways to increase the overall performances are: Deep SubMicron (DSM) technologies, new architectures, mono-core technologies with higher frequencies, multi-core technologies: COBHAM-Gaisler is getting close to the release of a SPARC multi-core Leon4-FT (the Next Generation Multi-Purpose Microprocessor, now GR740) and intended to be implemented on a DSM Technology (65 nm from STMicroelectronics). A dual-core LEON3-FT SPARC architecture fabricated in 180 nm CMOS technology and making use of Radiation-Hard-by-Design techniques and customized silicon technologies to ensure an adequate tolerance against radiation effect is already available by Aeroflex Gaisler (GR712RC) (Fig. 5).

For GR740 delivery of sample parts to ESA is planned in Q2/2016 with functional and radiation validation throughout 2016. After this, qualification should be started; however, no funding is available so far, waiting for a confirmed end user.

The potential of introducing a Time and Space Partitioning software architecture, has been inspired by the successful deployment of the Integrated Modular Architecture (IMA) in the Aviation industry (for instance, IMA is used on AIRBUS 380 and Boeing 777 airplanes). The advantages of the application of a Time & Space partitioning on the cost and development time of the Application SW is evident but its evident as well that the Hardware has to provide an environment that could ensure that each application cannot interfere with others, maintaining the isolation of the developed functional and processing chains (AOCS, Data Handling, TM/TC).

The specific interfaces can be implemented in another FPGA/ASIC (companion device) that could be a product of a specific OBC supplier while the core could be a generic component or Building Block. This solution has another technical advantage: it can take out from the high-performance processor some or all the low data

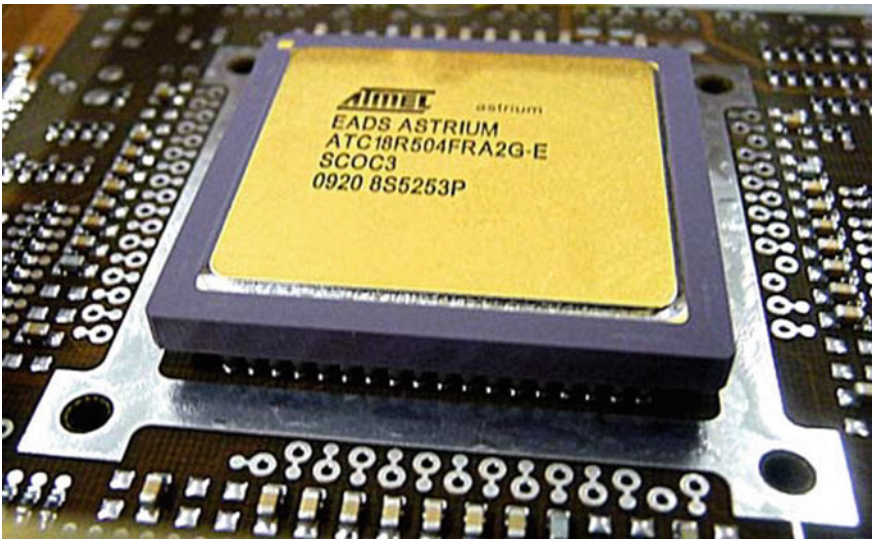


Fig. 5 SCOC3 ASIC (courtesy AIRBUS DEFENCE and SPACE)

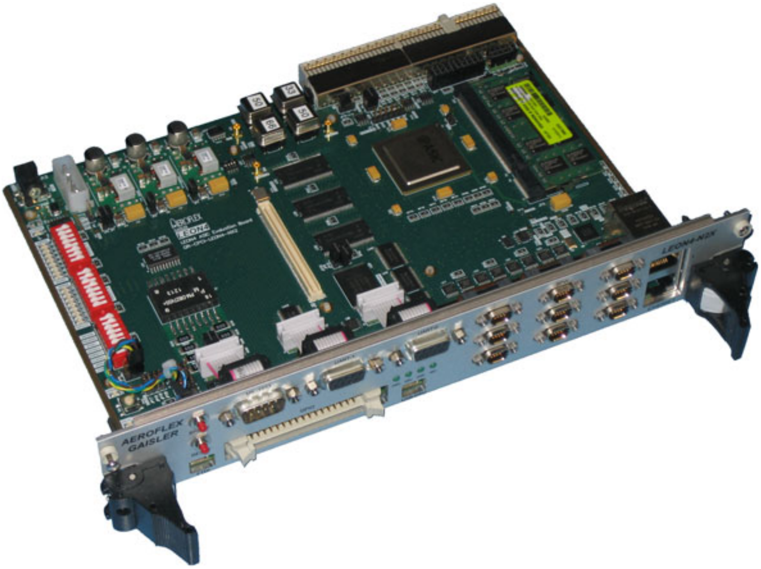


Fig. 6 GR740 development board (courtesy COBHAM-Gaisler)

rate interfaces (as UART or 1553 or CAN) or General-Purpose I/Os that with some penalties or with no evident reason are integrated in high-speed silicon platforms. Furthermore standard uC-based building blocks for interface management can be conceived (Fig. 6).

Bringing this solution to its extreme limits the core or the high-performance processor could become a sort of fast coprocessor used to run only some high-performance tasks and the centre of the scene is now occupied by the companion device that indeed could also operate with different type of coprocessors (e.g. also COTS parts).

It can be noted that although faster uProcessor are available (Leon 2/3) or will be available soon (e.g. NGMP) their exploitation is limited by the unavailability of suitable fast memories. In particular the Static RAM are slow and imposes to add wait states, the SDRAM are becoming obsolete and they can be affected by SEFI that requires complex protection/mitigation while other kinds of Dynamic RAM (e.g. DDRx) still to be fully certified to be tolerant to the space environment are supported only by the memory controllers embedded in the next LEON processor chips (GR740): fast memory development should also be supported and made available.

It is so extremely important to define a fast, reliable, standardized (non-proprietary) intra-processor link between the core and the companion chip that can guarantee a design lifetime in excess of 10 years, the availability of a fast and reliable interface is also a need for the Payload data processing.

The RTG4 FPGA has already set the standard for future inter-processor links with their embedded high-speed SERDES. The RTG4 devices have embedded high-speed SERDES blocks that can support data rates between 1 and 3.125 Gbps. The high-speed serial interface block supports several serial communication standards:

- XGXS/XAUI Extension (To implement a 10 Gbps XGMII Ethernet PHY interface);
- Native SERDES interface facilitates implementation of Serial RapidIO in FPGA Fabric or an SGMII interface to a Soft Ethernet MAC;
- PCI Express (PCIe) Gen1 Hard IP Core: x1, x2, x4 Lane(s) PCI Express Core; Up to 2 Kbytes Maximum Payload Size;
- 64-/32-bit AXI/AHB Master and Slave interfaces to the Application Layer.

However, the PCS logic can be implemented in the FPGA fabric, and the EPCS interface signals of the SERDES block can be connected to user protocol. This allows any user-defined high-speed serial protocol to be implemented in the RTG4 device.

In conjunction with EPCS, the available CorePCS IP module supports programmable 8B10B encoding and decoding. 8B10B is commonly used in protocols that are not included in the SERDES block by the high-speed SERDES interface. Therefore, the CorePCS IP module can be used with these protocols. It can be configured as a transmitter only, receiver only, or both transmitter and receiver. Word alignment support is included in the receiver. It can also be configured to support 10-bit or 20-bit EPCS data.

It should be noted that for small medium classes of satellite the split between On-Board Computer and On-Board Data Processing Unit does not apply anymore:

their integration has an immediate positive effect on mass, power, volume and cost, and is therefore an essential next step for Europe to maintain our leading role in optimized computer systems.

In terms of external interfaces that connect the OBC with the other elements of the DHS and with the Payloads, the situation is good in term of standardization 1. Europe has a quite complete list of standards that are covering the majority of the interfaces used on the Spacecrafts and launchers and the recent add on of ECSS-E-50-15C on CAN/CANOpen as well as the planned standardization of UARTs and SPI will further complete the picture. Activities to develop a SpaceWire-D protocol (where D stands for Deterministic) and Digital Sensor buses have also started and are well underway.

On the contrary the situation in term of availability of European products to implement these interfaces is not so positive but have improved somewhat since previous harmonization cycle. There are currently no European suppliers for RS422/485 and these interfaces are still largely used in all classes of spacecrafts.

The qualification process to have an European 1553B transceiver has not finished yet most of the designs still rely on US components.

In relation to LVDS transceivers (used by SpaceWire), activities have been funded by ESA under the ECI programme with the aim to have a European supplier of these key components: COBHAM (S) (formerly Aeroflex Gaisler) in cooperation with IMEC (B) are currently developing European alternatives of space-qualified dual transceiver and cross point switch components. LVDS components from STMicroelectronics (drivers, receivers, x-switch) are available and pending ESCC qualification. Also SPACE-IC has LVDS drivers and receivers at an advanced state of readiness. European components have to find their place in an high competitive market for LVDS, where several other US components have products (Texas Instruments, COBHAM).

During the past harmonization cycle, European industry failed in developing a solution for a 3.3 V ISO CAN transceiver. This allowed three US companies to take over what looks a promising and strategic market.

Another area of investigation in the Computer Architecture for next OBC is the internal power distribution scheme: low voltages as 3.3 Vdc but also 2.5, 1.8 and 1.2 Vdc and even less will be required with high current value capability, very high accuracy and possibility to implement redundancy scheme (usage of Point of Load converters seems mandatory for the future OBC and Mass Memory architectures). There are several available components (IR, Texas Instruments, JAXA, 3D+) and others are in development.

5.3 IP Hierarchy and Associated Values

More than half (in terms of value) of high-end consumer electronics shipped in the recent years were attached to reference designs supplied by key EEE component vendors, according to several market research studies. The emergence of the reference design programme by component suppliers such as Texas Instruments (that is pursuing the same strategy for Hi-Rel market too), MediaTek, Qualcomm, Renesas and Nvidia have helped small vendors, particularly in China and India, to compete effectively, for example, in the smartphone market, although at the beginning this strategy targeted mostly the lower cost devices (think about Xiaomi and OnePlus).

Reference designs are often used by designers to kick start new developments, to help them solve what they perceive as the biggest challenges in a design, and/or to evaluate design trade-offs, together with bringing their products to market faster. By having a significant part of the design complete, designers can then better evaluate what they need to bring the design to the necessary maturity.

Although tier-one electronic manufacturers have resisted delegating the device reference design to component suppliers because they see it as an integral part of their original know-how and differentiation, growing competition from small vendors is now forcing tier-one manufacturers to change their strategy and consider using third-party reference designs for cost-sensitive segments of the market: according to ABI Researches Mobile Device Semiconductors Market Research, Nokia/Microsoft, Samsung, HTC, LG, Huawei and ZTE have already started using this strategy for products targeted at emerging markets. These trends will be soon applicable in space and Hi-Rel businesses too, and take the competition to another level, forcing prime integrators and established unit manufacturers to make more compromises on reference design ownership. As a result, reference designs owned by component suppliers will gradually move to higher price points and added value, making the electronic products market increasingly commoditized. For smartphones (the most advanced electronic market around) ABI Research projects that more than two-thirds of smartphone shipments will be based on chipset suppliers reference designs by 2019, totaling more than 1.2 billion units. One quarter will be targeted at wholesale prices higher than 200 USD. The increase is expected to help chipset suppliers gain more clout in the mobile value chain and take the lead in smartphone technology innovation (Fig. 7).

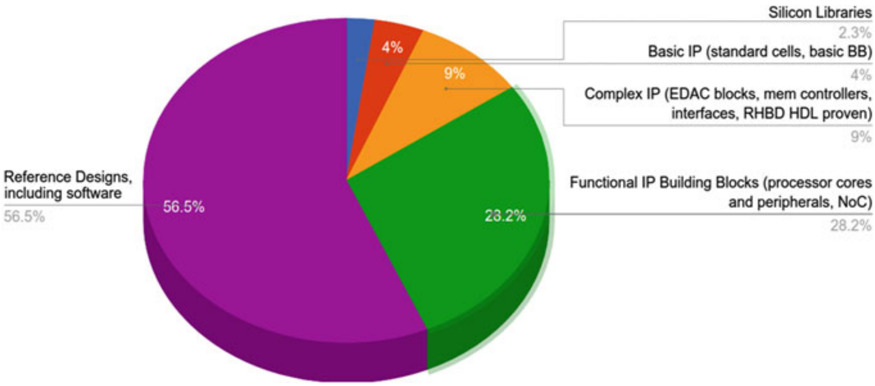


Fig. 7 Current commercial electronic market application value for silicon related IPs

A similar approach can be taken to stem growth and competitive environment in European Space Industry, the OBC market was very positively influenced by the availability of IP cores (ESA or Commercial) allowing fast prototyping and design of complex systems. Company know-how, rather than being diminished was boosted by the increased productivity and the portfolio of ASSP and design increased their market competitiveness, both at prime level (TAS, ASD example) and at equipment manufacturers (RUAG). There is the room to apply this industrial setup to many other fields beyond IPs. The availability of cheap distributed intelligence (in form of microcontrollers) allows for standalone design blocks for many avionics functions.

A trivial but relevant example is **Thermocouples and simple thermistors acquisition**: Temperature Sensors (Thermistors and Thermocouples), Pressure sensors, Position Sensors (Encoders, Resolvers, Contactless angular sensor based on Hall effect sensors), Accelerometers used in space applications have been acquired as analogue signals up to now. Potential problems with thermocouples (and also for many other resistor-based sensors like NTCs and PT1xx) include low level outputs, poor sensitivity and non-linearity. The low level output requires stable signal conditioning components and makes required system accuracy difficult to achieve. Connections in thermocouple systems must be made with great care to get good accuracy. Unintended thermocouple effects (e.g. solder and copper create a 3V/C thermocouple) in system connections make end-to-end system accuracies better than 5C difficult to achieve. A natural evolution is pushed by the need to increase the signal integrity and resolution of the transmitted signals and by the availability of miniaturized/mixed ASIC-sensors able to locally include the sensor biasing and signal conditioning/processing functionalities. The push towards an internal definition of standards for digital transmission of sensor data in spacecrafts, like I2C or SPI will only make this process easier. Currently there are no alternatives that do not need some sort of in-circuit pre-conditioning. Thermocouple Sensitivity is 10 to 100 uV/K thus need about 20bit ADC. One has to ensure that no current is taken from the Thermocouple, thus the ADC needs to have high-input impedance. Thus the need for a MOS to buffer the input signal. However, with MOS one get 1/f noise that has to be taken away either with CDS or a chopper technique (or other). With high resolution ADCs this should all be implemented in a standard, straightforward way. The only issue to check is whether the ADC input stage is a sampler with a capacitor or a gate to buffer the input signal. The availability of a better than 20 bit ADC that provides differential input impedance Chop off 100 G, Chop on 1 G and Common-mode input impedance 100 M, could be (more than) enough to do this job.

6 Radiation Risk Reduction for Small Spacecrafts and New Designers Using COTS Parts

Space can be an extremely challenging environment for electronic components and the reason for this is the presence of radiation particles. There are three main sources of radiation in the Earth vicinity: the Sun, which in addition to the solar wind can emits solar energetic particles (mainly protons but also heavier particles) during Solar Flares or Coronal Mass Ejection; the trapped electrons and protons in the radiation belts surroundings planets with a magnetic field (Earth, Jupiter, Saturn, Uranus) and finally the cosmic rays composed by particles and nuclei with very high energy which are generated inside or outside our galaxy by most likely supernova explosions.

Any space system (regardless of how cheaply built) shall respect several national and international regulations on space debris mitigation [10]. Thus a guaranteed reliability (and some knowledge of availability) is the major issue to face for space electronic systems. Any space system (except some manned spaceflights) should be considered as a self-standing intelligent unit, that is capable to tackle autonomously all the contingencies that happen during its lifetime.

The case for cubesats is compelling in this sense. Prof. Swartwout of St. Louis University painstakingly maintains a data base of mission status for the dozens of new smallsats sent every year [11]. Several related publications address the problem of reliability of smallsats [12, 13], but comprehensive data are missing to understand role of use of COTS with unknown radiation performance in it.

The on-board data handling system of a satellite is basically an embedded computer, but one made of rather different components compared to its terrestrial counterparts. Off-the-shelf computers and components do not last long in orbit. As cosmic rays or other high-energy particles pass through a spacecraft they can disrupt data systems by randomly flipping memory bits. These ‘single-event upsets’ are temporary in nature but reduce system’s availability and may trigger system level failures. A more serious radiation hazard still is a ‘single-event latch up/burnout’ when a charged particle causes a current surge that permanently burns out a chip or combined total ionizing dose (TID) effects, that put a limit on the operative lifetime of some components (For a complete view of Radiation harness Assurance on modern space systems consult [14]).

Varied mission life and complexity is growing for small spacecrafts that are now trending. Small missions benefit from detailed hazard definition and evaluation as done in the past, but requirements need to not overburden the designers that are often cost-aware (increased COTS usage).

Capturing the system impact of radiation device responses is tied into the verification of requirements and system performance. If only looked at from the piece part level these types of effects could impact availability, critical functions or mission success (see [15, 16] for a deeper insight).

6.1 Criteria for Selection of Parts for Enhanced Reliability

For Space vehicles or satellites in low inclination $(<28°)$ Low Earth Orbit (LEO), <500 Km) in both northern and southern hemispheres, typical dose rates due to trapped Van Allen electrons and protons are 100–1000 rad(Si)/year.

For Space vehicles or satellites in higher inclinations $(20 < I > 85^\circ)$ LEO in both northern and southern hemispheres, typical dose rates due to increased number of trapped electrons are 1–10 Krad(Si)/year.

There are three categories of components having the following characteristics:

- Commercial:
  - Process and Design limit the radiation hardness
  - No lot radiation controls
  - Total Dose: 2–10 krad (typical, lower technology nodes get better results)
  - SEU Threshold LET: $< 5\,\mathrm{MeV} \cdot \mathrm{cm}^2/\mathrm{mg}$
  - SEU Error Rate: 10^{-4} errors/bit-day (typical)
  - Latch-Up behaviour not guaranteed (Thresholds as low as $5\,\mathrm{MeV} \cdot \mathrm{cm}^2/\mathrm{mg}$ possible)
  - Customer performs rad testing, and assumes all risk
  - Customer evaluation and risk.
- Rad Tolerant:
  - Design assures rad hardness up to a certain level
  - No lot radiation controls
  - Total Dose: 20–50 krad (typical)
  - SEU Threshold LET: $20\ \mathrm{MeV} \cdot \mathrm{cm}^2/\mathrm{mg}$
  - SEU Error Rate: 10^{-7}–10^{-8} errors/bit-day
  - Latch-Up behaviour tested (Thresholds known, typically above “Iron Knee”)
  - Usually tested for functional fail only, risky
  - Customer evaluation and risk.
- Rad Hard:
  - Designed and processed for particular hardness level
  - Wafer lot radiation tested
  - Total Dose: $> 100\,\mathrm{krad}$ to $> 1\,\mathrm{M\,rad}$
  - SEU Threshold LET: $> 60\,\mathrm{MeV} \cdot \mathrm{cm}^2/\mathrm{mg}$ (impossible for memories)
  - SEU Error Rate: 10^{-10} to 10^{-12} errors/bit-day
  - Latch up: None.

Using rad-soft components does not significantly reduce cost, but greatly increases risk. There are no components that are ideal for all parameters. Electronics and integrated circuit’s design requires many trade-offs in performance (cost included as performance parameter), so commercial components are useful only for commercial applications, where low cost, latest technology (even if it is immature) and high speed takes precedence over extreme temperatures and voltage ranges. Shielding these devices in Space Applications is a futile effort, especially for Single-Event Effects such as SEU and Single-Event Latch up (SEL). Nevertheless, as discussed before, quick push towards mass-produced space electronics is opening the window towards ‘functionally safe’ architectures that allow safer use of commercial electronics with guaranteed availability and reliability performances.

7 Conclusions and Market Perspectives

Command & control and data processing spacecraft electronics comprises a variety of products corresponding to the variety of functions needed on a satellite: computers, data processing equipment, mass memories, command/drive electronics, instrument front-ends, atomic clocks, etc. Competition on quality, price and timeliness is still concentrated among a limited number of suppliers despite more equipment manufacturers entering the space industry.

To assess this market we consider that the contribution of electronics for launchers is very small compared with the electronics embarked in satellites. From published generic parametric cost estimates we assume that the average cost of command and data processing electronics (platform and payload) is about 12% of the total satellite cost.

During the last five years, the European industry has been selling satellites by an averaged approximate amount of €5B per year (launchers approximately €2B) equivalent to more than 35% of the world market. On these basis a development and manufacturing of data systems electronics in Europe of about €600 M per year can be assumed. The European space industry realizes more than 40% of its consolidated sales with commercial customers (understood as commercial operators and non-European governments) so it is well prepared to have the same share of the foreseen open market of ~€20B/year for the next decade.

After few years of stability at around €5 billion of annual sales of European space industry, space sales reached a record €7.2 billion in 2014, mostly driven by sales to European institutions and foreign governments. According to ASD-Eurospace, commercial sales of the European space industry accounted for 46% of total sales, or €3.3 billion.

Continuous R&D investment and a skilled labour force are prerequisites for the established satellite industries to continue to deliver reliable and cost-effective satellite systems. The European and Japanese space industries, unlike their U.S. counterpart, do not have access to large domestic military markets. They have yet to find alternative sources of financing for dual-use R&D and for flight qualifications of new technologies.

Governments will remain the largest customer of the satellite and launch service industries in the next 10 years, with 860 satellites to be launched from 2015 to 2024 (i.e. a 32% decade-to-decade increase). The governmental dominance of the world space market will remain true even if the two commercial mega-constellations of small comsats are launched (OneWeb and Steam). The two constellations together represent capex of USD 13.5 billion for 5000 satellites; the 860 government satellites have a market value of USD 192 billion.

So, accounting for the uncertainties of the current constellation projects, the most likely scenario (Euroconsult Data) is that in the next 10 years will be produced 1.410 commercial and institutional satellites valuated €200B (€20B per year). The command and data processing electronics installed in these 1.410 satellites will generate an industrial turnover of approximately €2.5B per year worldwide.

Assuming the European market share will remain the same in the next decade compared to the past five years, the European satellites will generate industrial activity reaching approximately €850 M yearly average, related to development and manufacturing of command and data processing electronics units. The average cost of electronic components in command and data processing units is about 35% of the sales price of these units, leading to electronic parts procurement for 300 M€/year by EU manufacturers.

As for many hi-value parts (e.g. FPGAs, microcontrollers, processors, RH power systems) US manufacturers still dominate the market, thus an increased level of R&D activities is needed to support a stronger EU return in this field.

References

Material for this chapter has been re-edited and adapted for public use from several internal European Space Agency's reports and documents. For any detailed references on ESA's programs please consult ESA's website at http://www.esa.int.

1. ESA dedicated support for development of mega-constellations, https://artes.esa.int/news/esa-announces-dedicated-support-development-megaconstellations
2. M. Fabiano, G. Furano, NAND flash storage technology for mission-critical space applications. IEEE Aerosp. Electron. Syst. Mag. **28**(9), 30–36 (2013)
3. Space avionics open interface architecture initiative, http://savoir.estec.esa.int/
4. Cobham Gaisler is the main providers for IP cores and tools for LEON SPARC architecture, http://www.gaisler.com
5. The Lexra Story by former employee Jonah Probell, http://probell.com/lexra/
6. G. Furano, J. Ilstadt, R. Jansen, G. Magistrati, K. Marinis, D. Merodio, M. Rovatti, *in support of a FPGA criticality defined validation, with particular focus on radiation effects*, vol 720 (ESA Special Publication, 2013), p. 25
7. The first-ever Rad Tolerant megaAVR, http://blog.atmel.com/2015/10/21/the-first-ever-rad-tolerant-megaavr-is-out-of-this-world/
8. SCOC3: a space computer on a chip. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010, pp. 1345–1348
9. Spacecraft Controller On a Chip (SCOC3), FINAL REPORT (2013), http://microelectronics.esa.int/finalreport/SCOC3-ExecutiveSummary-I01R00.pdf
10. Compendium of space debris mitigation standards adopted by States and international organizations, http://www.unoosa.org/oosa/en/ourwork/topics/space-debris/compendium.html
11. CubeSat Database, https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database
12. M.A. Swartwout, CubeSats and mission success: a look at the numbers, presented at the 2016 CubeSat Developers Workshop, San Luis Obispo, April 2016
13. M.A. Swartwout, CubeSats: toys, tools or debris cloud? invited talk at the 2014 St. Louis Space Frontier Gateway to Space Conference, 8 November 2014
14. J.R. Schwank, M.R. Shaneyfelt, P.E. Dodd, *Radiation hardness assurance testing of microelectronic devices and integrated circuits: radiation environments, physical mechanisms, and foundations for hardness assurance*
15. Review of radiation hard electronics activities at European Space Agency. J. Instrumen. **8–2** (2013), IOP Publishing
16. D. Sinclair, J. Dyer, *Radiation effects and COTS parts in SmallSats*, 27th Annual AIAA/USU Conference on Small Satellites, Logan, Utah, 2013