Universitat Politècnica de Catalunya

Cache Memory Design in the FinFET Era

Student: Zoran Jakšić
Director: Ramon Canal Corretger

Acta de calificación de tesis doctoral

Curso académico:

Nombre y apellidos

Programa de doctorado

Unidad estructural responsable del programa

Resolución del Tribunal

Reunido el Tribunal designado a tal efecto, el doctorando / la doctoranda expone el tema de la su tesis doctoral titulada _____
_____.

Acabada la lectura y después de dar respuesta a las cuestiones formuladas por los miembros titulares del tribunal, éste otorga la calificación:

☐ NO APTO ☐ APROBADO ☐ NOTABLE ☐ SOBRESALIENTE

(Nombre, apellidos y firma) Presidente/a	(Nombre, apellidos y firma) Secretario/a

(Nombre, apellidos y firma) Vocal	(Nombre, apellidos y firma) Vocal	(Nombre, apellidos y firma) Vocal

_____, _____ de _____ de _____

El resultado del escrutinio de los votos emitidos por los miembros titulares del tribunal, efectuado por la Escuela de Doctorado, a instancia de la Comisión de Doctorado de la UPC, otorga la MENCIÓN CUM LAUDE:

☐ SÍ ☐ NO

(Nombre, apellidos y firma) Presidente de la Comisión Permanente de la Escuela de Doctorado	(Nombre, apellidos y firma) Secretario de la Comisión Permanente de la Escuela de Doctorado

Barcelona a _____ de _____ de _____

Cache Memory Design in the FinFET Era

Zoran Jakšić
Department of Computer Architecture
Universitat Politècnica de Catalunya

A thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

Advisor:
Ramon Canal Corretger
Universitat Politècnica de Catalunya

May 2015

Acknowledgements

Doing a PhD is certainly a very important thing in the life of every individual who decides to start that journey at some point in their life. Although the reasons may be different, I believe that the most of students starts it from one of these two; a childish love for research and desire to work on an interesting problem, and egoistic wish to put a PhD title in front of their name. I started my doctorate from both.

However, very soon after you start that job, you realise that research (as well as life) is not a simple task. It is not a high-school math assignment that you know that has exactly one solution that you have to find in a predefined time. It is rather some floating in the unknown where, at first, you have to realise what kind problem do you have, and then try to solve it even if you do not know does the solution exist. In that environment you are faced with different situations, often for the first time, and if you are not mature and trained very well you pass to the numerous psychological fazes spanning from the high motivation and enthusiasm for making something right to the deep depression with low self-esteem and disappointment.

Traveling that journey of life and research, I believe that I have got to know myself better, and among all I have realised the importance and the meaning that different people had in my life and this PhD. On these pages, I would like to thank all of them who helped me, taught me the valuable life lessons, or simply make me feel happy during these four years of my studies.

The most gratitude for completing this thesis I owe to my supervisor Ramon Canal. I hope that I will not sound too much like speaking empty phrases when I say that having good advisor is the most important part of every successful doctorate. Having the supervisor that gives you the actual topic, great liberty to work and develop your own ideas but on the same time always being available to discuss those ideas critically, tells you when you have enough material to start writing a paper and helps you with it in the best possible manner is indispensable. Doing all that with constant encouragement that a candidate is doing a good job is crucial for keeping the motivation of a student. Ramon indeed was an excellent supervisor in all these aspects, and moreover a friend from the first day I came to UPC.

Besides Ramon, I would like to give my thanks:

To the professor Antonio Gonzalez, director of the ARCO group, for giving me the opportunity to do this research. From him, I have learned how important is presenting the work, knowing to ask a precise question and to answer clearly. Possessing the vast expertise in the field and knowing to express it in a simple and understandable manner is the most important quality that I truly respect in Professor Antonio Gonzalez, and I am really thankful for the chance to listen him and to learn from him.

To Shrikanth, for helping me with the simulation environment when I came at UPC and for his errorless organisation of our joint trips to the conferences.

To Daśa, for suggesting me to apply for the PhD position at ARCO group. Without his suggestion, nothing of this would happen. I hope that one day I will prove him that my PhD is not 1% my and Ramons work and 99% his suggestion me to apply.

To Čedomorac a.k.a. Batica, the earliest childhood friend, for always being available for the chess game to short my day in moments when I lacked a motivation for the work.

To Bokule for being always available for a tea break and a friendly chat in the FIB cafe, and also for paying me the tea there once. And to Djomla for being excellent replacement for the tea breaks when Bokule left.

To Marc, for being the best flatmate that I have ever had because I have been seeing him almost never in the flat. And to Lošmi who in spite of his good cooking and being a great friend, was the worst flatmate I have ever had; but who because of that, thought me a great patience, tolerance, self-control and, eventually made me a better person.

To Panović for a kindly explaining me how to use an oven and finally prepare a normal meal.

To Rajili, for always reminding me to my home country and the glorious history of Serbian people (especially during the World War I) that I should be proud off. And, no Rajo, this does not mean that I will "donate" anytime soon.

To Radule, for the weekend runnings, and to Uglješa for a try to run with me once although he quit after the first kilometre.

To Ješka, for being the only friend always capable of listening my life philosophy without ever saying me "Oh, shut up, please".

To Nidžo and Mišur, two home friends who always prepare me the excellent program of doing nothing when I go back on holiday in Montenegro.

To Bane and Zlajo, for showing me on their own anti-example that for achieving success it is not enough just to wish to do something. One have to work to achieve his goals, too.

To Branimir and Martina, two brilliant astrologers who predicted me earning a big money at some point in my life. I really believe this is going to happen because it is impossible that some astrologer miss every possible thing, and they did about me.

To Mennan, Manos, Vamis and Nav, for the improvisation of the table tennis in the office. Thank God this came at the final stage of my PhD because who knows when I would have finished it.

To Oscar, for being the only quiet person in the office when present I could actually work something.

To Manish, for putting some money in my pocket and for (I hope) one day he will explain me that joke that leaves all people confused.

To my sisters Jelena and Milica for... Well, everyone thanks to the family so let put them here without any particular reasons. I hope that one day, I will figure out their concrete merits in my life.

To my mother, for being the only person in the world who truly believes that I am a genius.

And finally, to the committee members, for the useful comments and for (I hope) not being so strong with me on my pre-defence and defence. After all

Nobody expects the Spanish Inquisition

Monty Python's Flying Circus
Spanish Inquisition Sketches

Dedicated:

To the memory of my father, who told me once that I should do a PhD because every son should reach further than his father.

To my mother, for her unconditional love and support during all these years.

Posvećeno:

Sećanju na moga oca, koji mi je davno rekao da bih trebao da zavrsim doktorat jer svaki sin treba da stigne dalje od svoga oca.

Mojoj majci, za bezuslovnu ljubav i podršku svih ovih godina.

Abstract

The major problem in the future technology scaling is the variations in process parameters that are interpreted as imperfections in the development process. Moreover, devices are more sensitive to the environmental changes of temperature and supply voltage as well as to ageing. All these influences are manifested in the integrated circuits as increased power consumption, reduced maximal operating frequency and increased number of failures.

These effects have been partially overcome with the introduction of the FinFET technology which have solved the problem of variability caused by Random Dopant Fluctuations. However, in the next ten years channel length is projected to shrink to 10nm where the variability source generated by Line Edge Roughness will dominate, and its effects on the threshold voltage variations will become critical.

The embedded memories with their cells as the basic building unit are the most prone to these effects due to their the smallest dimensions. Because of that, memories should be designed with particular care in order to make possible further technology scaling.

This thesis explores upcoming 10nm FinFETs and the existing issues in the cache memory design with this technology. Moreover, it tries to present some original and novel techniques on the different level of design abstraction for mitigating the effects of process and environmental variability.

At first original method for simulating variability of Tri-Gate FinFETs is presented using conventional HSPICE simulation envi-

ronment and BSIM-CMG model cards. When that is accomplished, thorough characterisation of traditional SRAM cell circuits (6T and 8T) is performed. Possibility of using Independent Gate FinFETs for increasing cell stability has been explored, also.

Gain Cells appeared in the recent past as an attractive alternative for in the cache memory design. This thesis partially explores this idea by presenting and performing detailed circuit analysis of the dynamic 3T gain cell for 10nm FinFETs.

At the top of this work, thesis shows one micro-architecture optimisation of high-speed cache when it is implemented by 3T gain cells. We show how the cache coherency states can be used in order to reduce refresh energy of the memory as well as reduce memory ageing.

Contents

Glossary

C_{OX} Gate Capacitance.

L_{eff} Effective Channel Length.

T_{fin} Fin Thickness.

V_{TH} Threshold Voltage.

μ Carrier Mobility.

σ Standard Deviation.

q Electron Charge.

ABB Adaptive Body Biasing.

ASB Adaptive Source Biasing.

ASV Adaptively Scaling the Supply Voltage.

BIST Bilt In Self Test.

BSIM-CMG Berkeley Short-channel IGFET Model - Common Multi-Gate.

BTI Bias Temperature Instability.

CMP Chemical Mechanical Polishing.

DRAM Dynamic Random Access Memory.

DVFS Dynamic Voltage and Frequency Scaling.

ECC Error Correction Codes.

EDC Error Detection Codes.

EOT Efective Oxide Thickness.

FBB Forward Body Biasing.

FER Fin Edge Roughness.

FinFET Fin Field Effect Transistor.

GER Gate Edge Roughness.

HCI Hot Carrier Injection.

HSPICE Simulation Program With Integrated Circuit Emphasis.

IC Integrated Circuit.

IG Independent Gate.

ITC Interface Trap Charge.

ITRS International Technology Roadmap for Semiconductors.

L Chanel Length.

LER Line Edge Roughness.

MESI Modified Exclusive Shared Invalid, Cache Coherence Protocol.

MGG Metal Grain Granularity.

MOESI Modified Owned Exclusive Shared Invalid, Cache Coherence Protocol.

NBTI Negative Bias Temperature Instability.

NM Noise Margin.

PBTI Positive Bias Temperature Instability.

PHIG Gate Work-funtion.

RAT Read Access Time.

RBB Reverse Body Biasing.

RD Reaction Diffusion.

RDF Random Dopant Fluctuations.

RMS Root Mean Square.

RSNM Read Static Noise Margin.

SOI Silicon On Insulator.

SRAM Static Random Access Memory.

TCAD Technology Computer Aid Design.

TG Tri-Gate.

W Channel Width.

WLWM World Line Write Margin.

List of Figures

List of Tables

Look, matey, I know a dead parrot when I see one, and I'm looking at one right now.

Monty Python, Dead Parrot

1

Introduction

Humanity has witnessed tremendous progress since John Bardeen, William Shockley and Walter Brattain invention of the transistor, and Jack Kilby's first assembly of an integrated circuit in 1948 and 1958 respectively.

The exponential increase of the transistor number on a die (and by that, circuit performance) followed by the famous Moore's law was the main reason of technology advancements for almost 50 years. Integrated Circuit (IC) development followed almost perfectly this famous law that predicts reducing the transistor area by half approximately every 18 months. As a consequence, the number of transistors has increased exponentially from the 2300 that had the Intel 4004, (the first commercial microprocessor released in 1973), to the 4.3 billions (that can be found in Intel's 15-Core Xeon Ivy Bridge-EX processor, the biggest microprocessor that can be found on the market at the moment of writing this document).

However, the time of simple scaling of transistor features came to the critical point as the dimensions felt below 32nm because imperfections in the development process could not be neglected anymore. Simple technology scaling of the planar devices is more complicated with every generation as the transistor channel length is approaching the critical 10nm. Large companies

invest a lot more funds in research that should prolong transistor scaling that, in turn, increases the transistor cost. According to one definition of Moore's law that assumes doubled performance for the same price every 18 months; this law will be dead in a couple of years (if the high volume and the vast diversity of the devices does not lower the transistor price).

It is not rare to read articles that talk about devices that we can expect in the post-silicon era. Some of them discuss even revolutionary technologies like DNA computing or even quantum computing. However, although these technologies, in theory, seem very powerful, significant time is going to pass before any commercial application is made. In other words, semiconductor technology is still going to be very present in the next 10-15 years and scaling the transistor features is going to be the primary reason of the technology advancement.

Dead or not, Moore's law is at its critical point, and the time of simple transistor scaling is behind us. New solutions and approaches in how we design chips are necessary for all of the levels of design abstraction if we want to see further technology advancement and performance improvement.

1.1 Challenges in Further Technology Scaling

In the current stage of silicon technology manufacturing, process variations pose a critical limiting factor. Variations in process parameters can be understood as imperfections in the development process which translate to a non-deterministic behaviour of the transistors. These imperfections do not scale with technology; thus their effect on circuit functionality increases as technology shrinks. As a consequence, transistors cannot be observed as deterministic devices in the design process. They rather have to be found statistically with parameters that follow some random variable distribution (most commonly a Gaussian Normal Distribution).

Besides the fabrication process, devices also change their characteristics over time due to temporal fluctuation of voltage and temperature. According to International Technology Roadmap for Semiconductors (ITRS) [1]; in the next 10-15 years, the nominal supply voltage of the semiconductor circuits

is going to drop down to 0.7V for high-performance devices. Consequently, the reliability of such devices is significantly challenged. Noise Margin (NM) which is the maximal voltage that can appear in some place in the circuit without causing an error in the logic, reduces for smaller voltage supply.

Temperature effects are expressed the most in terms of static power consumption. Traditionally, over the years IC designers were focused on minimising dynamic power since it was the dominant component in the overall energy budget of the devices. However, due to the high number of transistors on a die, the contribution of static power cannot be neglected anymore. Following the fact that leakage currents are exponentially dependent on the device temperature, the effects of temperature on the total power consumption of the chip are evident.

Besides the process and temporal variations, devices lose their characteristics due to ageing (Bias Temperature Instability (BTI), Hot Carrier Injection (HCI)) [2]. Over the time, silicon traps are accumulated on the Silicon-Oxide border. This physical process can be understood as an increase of the threshold voltage of the transistors that in the end translates in the lower driving current and consequently lower switching speed.

All in all, process and environmental variations pose significant challenges, and it is of crucial importance to consider their effects during design process [3]. Additionally, innovations at every level of design of the design abstractions are necessary.

Not so long time ago, Intel Corporation announced that it would use Tri-Gate FinFETs for implementation of their 22nm processors. That has put this technology as the definite successor of the classical bulk devices. Because of their 3D structure, they achieve better channel control and higher I_{on}/I_{off} is reached with lightly doped channels [4, 5]. Lightly doped channels reduce the effect of Random Dopant Fluctuations (RDF) which is the greatest source of variability in classical bulk technology. However, some sources of variability caused by Line Edge Roughness (LER) and Metal Grain Granularity (MGG) still remain, and their effects on the circuit performance is going to be greater as the device channel length approaches 10nm [6, 7, 8]. In other words, FinFET technology has solved the problem of process vari-

ability caused by RDF but many other are still left open, and they demand original solutions in order to mitigate them.

1.2 Embedded Memory Design in Presence of Process and Environmental Variability

Novel multi-core processor architectures demand more on-chip caches for efficient sharing information across parallel processing units. Traditionally, on-chip memories are based on the 6T SRAM cells. Since memory structures occupy the biggest part of the chip area, it is of crucial importance that these cells are scaled to the minimal dimensions. Parameter fluctuations have the greatest influence on circuits that rely on perfectly matched transistor pairs [8]. When the small dimensions of the devices are added on top of that it is very clear why SRAM cells are the most vulnerable to the effects of process variability. The variability of the devices in the embedded memories manifests as the increased static power consumption and the increased number of faulty cells.

Due to the high area that they occupy, they become significant consumer of static energy. One way, to fight this inflation, is to reduce supply voltage - V_{DD}. However, this reduction goes to the cost of the cell stability (the number of failures increases), and it cannot be accomplished below some critical value defined as $V_{DD_{min}}$. Also, supply voltage reduction decreases a circuit speed that is directly dependent on V_{DD}.

Due to these challenges, innovations in the cell design have to be found on the circuit level by presenting novel memory SRAM cells (e.g. 8T), or even change in cache memory approach altogether by adopting DRAM technology in high-speed cache memories. The second idea introduces another problem. These cells lose their state over time, and they have to be refreshed which increases the energy budget and produces performance losses because the memory is blocked during that process.

1.3 Thesis Contributions

The primary focus of this thesis is on FinFET technology and the characterisation of cache memories when they are implemented by these devices. Due to its geometry, the simulation of FinFET devices demands certain innovations in the simulation approach especially when process variability is introduced. Traditional, analytical models have to be adjusted, and optimal solutions for simulating variability have to be found. Besides that, the evaluation of traditional embedded memory circuits has to be completed in order to get an insight into the critical points and challenges. Finally, original solutions on the circuit and micro-architecture levels, unique to the available technology should be presented to find optimal solutions.

This thesis spans through different levels of design abstraction in the attempts to solve the major challenges in embedded memory design in the present days. In short, the main contribution of the thesis are:

- *Characterisation of the upcoming 10nm SOI FinFETs.* In this part of the thesis, we investigate how the process variation can be successfully incorporated into the standard HSPICE BSIM-CMG model for efficient circuit simulation using traditional Monte Carlo methods. It also presents how the effects of a back-gate biasing (that can be applied in Independent Gate (IG) FinFETs) can be simulated with Tri-Gate (TG) model card that was available for this research.

- *Characterisation of SRAM cells for the 10nm SOI FinFET technology.* This part consists of an extensive simulation of the traditional 6T and 8T SRAM cells when they are exposed to the effects of process and environmental variations. We simulated cell stability (Read Static Noise Margin (RSNM), and World Line Write Margin (WLWM)), as well as cell Read Access Time (RAT) and Static Power Consumption. We show how the RSNM and WLWM of the 6T SRAM cell can be improved when the back-gate biasing technique is applied in the cell design. Finally, we also compare to previously published work on the

SRAM cells in higher technology nodes to provide an insight into the scaling trends of FinFET SRAMs.

- *Dynamic 3T gain cell characterisation for the 10nm FinFET technology.* In this part, we investigate the possibility of implementing high-speed caches with the 3T gain cell. Thorough characterisation of the dynamic Gain Cell is presented, and results are compared with the classical 6T SRAM cell. We further enhance the cell retention time and consider the effects they have on a large memory array implementation. Finally, we compare our work with the similar solutions regarding DRAM cells and explain the advantages and disadvantages of our proposal.

- *DRAM based high-speed coherent caches.* In this part, we propose usage of the 3T dynamic cell for high-speed coherent caches. We evaluate what effects such implementation has on cache energy and performance. We further propose how the cache coherency can be exploited for reducing refresh energy by refreshing only the lines in certain coherence states and invalidating the other when they expired. This technique, also reduces the memory ageing, and we evaluate the benefits in terms of ageing reduction.

You see, your cat is suffering
from what we vets haven't found
a word for.

Monty Python, Confuse A Cat

2

Background

2.1 Introduction

In this chapter, we present current technology scaling challenges. At first, we explain more closely why variation exists and what are the major sources of variation.

Later, we present some conventional techniques that can be used to combat process variation at different levels of design abstraction. We discuss traditional techniques that can be applied on the circuit level of design (e.g. Body Biasing, Source Biasing). As the primary goal of this thesis is cache memories design, we discuss techniques that are particular for the memory design (cell modification and cell resizing). We further mention some conventional micro-architecture techniques for combating variability in cache memories.

2.2 Process and Environmental Variations

In the first section, we discuss process variations. We explain how these variations are classified into subgroups: random or systematic (within-die and

die-to-die). From the whole set of the physical sources of process variations, in this chapter we explain the most dominant two - Random Dopant Fluctuations (RDF) and Line Edge Roughness (LER). Some authors, (e.g. in [9]) add to this list L_{eff} but since this variation is the direct consequence of LER in sub 22nm nodes, we will exclude it here.

Second part is reserved for the environmental variations. We discuss the most common sources, voltage and temperature. We explain why the voltage variations exist between different parts of the chip as well as how the activity factor of different cores affects unbalanced heating of the chip.

Finally, we discuss what caused device ageing and how it affects circuit performance. As the most common source of ageing, we identified Bias Temperature Instability (BTI). We present the most acceptable explanation of this physical process (Reaction-Diffusion (RD) method) and major factors that influence it.

2.2.1 Process Variations

Process Variations can be understood as imperfections in the manufacturing process, which result in the non-deterministic behaviour of the transistors. In general, they can be classified as statistical or systematic.

Statistical variations exist due to the development process that is hard to control when device dimensions are very small (oxidation, etching, diffusion, ion implantation...). For instance, if the channel length is large enough, the number of dopants in the channel is very high. Accordingly, the error in their total number is small if just a couple of atoms are not ideally placed. On the other hand, if the channel is very short, and the total number of dopants, is extremely small, every single atom that deviates from the projected design produces the error that cannot be neglected. For instance, in 10nm technologies, the expected number of dopants in the channel is in the order of 100 atoms. These variations are totally random, and there is no better solution than to observe them statistically.

Another type of process variations can be classified as systematic. These variations deviated from their nominal projected value, but they are char-

acterised by the correlation between devices. They depend on the space and distinct patterns. An example of systematic variations is silicon surface flatness variations caused by the layout density variation during Chemical Mechanical Polishing (CMP) [9].

Systematic (or Spatial) variations, as they can be defined, can be classified into the two subgroups.

- *Intra-Die (Within-Die)* - These differences exist between elements on the same chip. They exist due to the layout geometry when the variation in a certain component causes a change in a nearly part. As the distance increases, the correlation between these variations reduces.

- *Inter-Die (Die-to-Die)* - Variations between chips in the same wafer or between different wafers.

2.2.1.1 Random Dopant Fluctuations

As it name suggests, this type of variations is caused by the imperfection in the development process resulting from the random number of dopant atoms in the channel. With the reduction of the dimensions of the devices with every generation, it is much harder to control the total number of dopants in the channel. This variation in the number of dopants in the transistor channel results in the variation of the threshold voltage (V_{TH}) for the device [10]. Figure 2.1 illustrates the effects of the random dopant fluctuations. Every small dot present one dopant atom. It can be observed that this number is very small comparing to the channel dimensions.

The negative impact of variations due to RDF was the reason for the introduction of FinFET technology (Chapter 3). In [11], authors show that variation of threshold voltage (ΔV_{TH}) for Tri-Gate FinFETs due to RDF can be calculated according to the Equation 2.1. ΔV_{TH} is the variation of threshold voltage, q is electron charge, T_{fin} is the fin thickness, T_{fin} fin height, L channel length and C_{OX} gate capacitance.

$$\Delta V_{TH} = \frac{qT_{fin}}{2C_{OX}} \sqrt{\frac{N_{ch}}{LT_{fin}H_{fin}}} \tag{2.1}$$

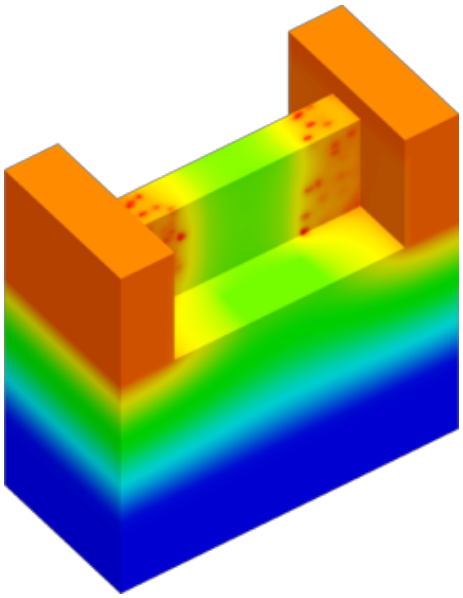


Figure 2.1: FinFET Random Dopant Fluctuations (RDF); Source Wang et. all [6]

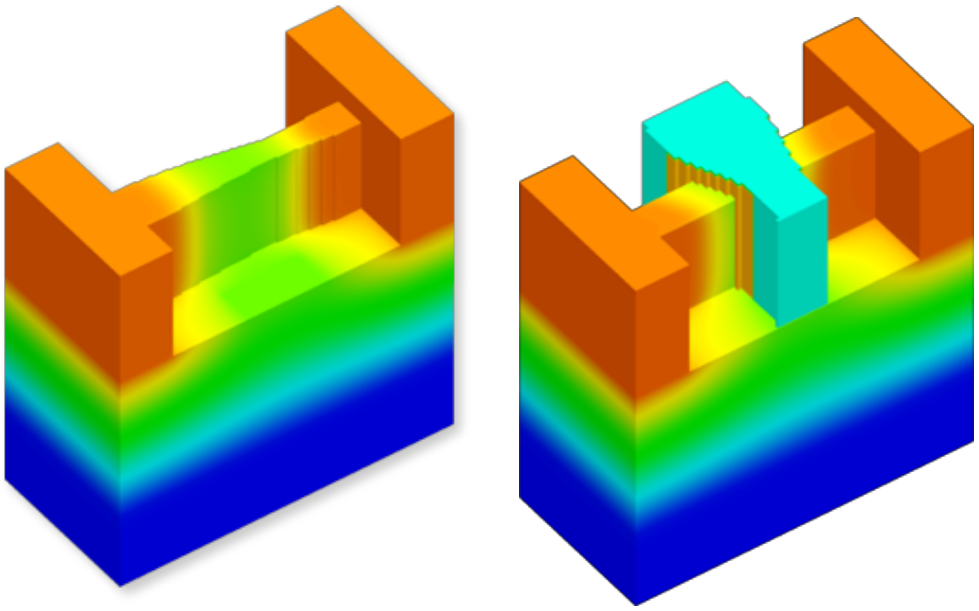
When the device parameters from the Table 3.1 are included in the Equation 2.1 it can be seen that for the 10nm FinFET technology, that we introduce in the Chapter 3, V_{TH} variation can be neglected.

2.2.1.2 Line Edge Roughness

Gate patterning introduces a non-ideal gate edge; this imperfection is referred as Line Edge Roughness (LER). As device scaling continues into sub 20nm regime, LER is expected to be a significant source of variation [6]. LER shows up since the nanometer technologies use light sources with wavelengths that are much larger than the minimal feature size. Besides that, LER is caused by a number of statistically fluctuating effects at these small dimensions such as shot noise (photon flux variations), statistical distributions of the chemical species in the resist such as photoacid generators, the random walk nature of acid diffusion during chemical amplification, and the nonzero size of resist polymers being dissolved during development. It is unclear which process or processes dominate in their contribution to LER [12].

For FinFET technologies, LER can be observed as Gate Edge Roughness

(GER) and Fin Edge Roughness (FER). This is illustrated on Figure 2.2



(a) Fin Edge Roughness (FER) (b) Gate Edge Roughness (GER)

Figure 2.2: FinFET Line Edge Roughness; Source Wang et. all [6]

These features are almost constant through the technology nodes, so their effects on the smaller feature size are much higher. According to [6], variations caused by LER is going to dominate threshold voltage variation for the 10nm SOI FinFET technology.

The standard deviation of LER-induced V_{TH} variation when devices dimensions are changing from W_1 to W_2 can be calculated according to 2.2 [12].

$$\sigma_{V_{TH}|W_2} = \sqrt{W_1/W_2}\sigma_{V_{TH}|W_1} \tag{2.2}$$

2.2.2 Environmental Variations

The electrical behaviour of the devices in a circuit strongly relies on the operating conditions. The current through a device is strongly dependent on the voltage at its nodes, and according to that to the supply voltage. The carrier mobility (μ), and consequently the threshold voltage (V_{TH}) are dependent

on the operating temperature, too. In addition to the process variations described in the previous section, any change in these environmental conditions, also causes power and performance deviations from their nominal projected values.

2.2.2.1 Supply Voltage Variations

The supply voltage at any location on a die deviates from the nominal design value depending on the current requirements of the design. Increasing the number of the devices in a die, and the operating frequency have augmented the current demands.

Various blocks in a design usually share a single power grid. Any sudden increase in the switching activity of a particular block requires additional current to be supplied to that area of the chip. The parasitic inductance in the power grid network and the package causes a di/dt drop and hence a transient voltage drop on the supply lines. Further, in multi-core processor architectures, dynamic changes in the current requirements of any one of the cores can create supply voltage variations on all of the others [12].

Moreover, wire properties are worsening with every generation which makes the problem of voltage scaling even more significant. Voltage drops across the power grid due to the resistance of the power supply lines, which is a function of the current. This is known as the IR drop and, it is proportional to the steady-state current of the power supply and the resistance of the power grid. The continued scaling of wire dimensions to match the device scaling has increased the resistance per unit length of a wire. In addition, growing die sizes with larger transistor counts (especially on-die caches) has increased the steady state (leakage) currents exacerbating the IR drop problem. The resistance of the wires also increases with temperature, which further worsens the IR drop [12].

2.2.2.2 Temperature Variations

For a given power density, the silicon temperature is a function of the thermal conductivity of the materials used. In a bulk CMOS technology, the gener-

ated heat spreads through the silicon substrate as well as the wires. However, in a silicon-on-insulator (SOI) technology the poor thermal conductivity of the buried oxide causes most of the heat to be carried away primarily along the wires, which causes the temperature to increase at a faster rate.

Another factor that determines the spatial temperature variation is the actual application run on the system. Individual blocks are continuously used in some workloads. In multi-core processors, it is likely that some of the cores are inactive and hence cool down for a greater percentage of time. Depending on the ability of the system to dynamically manage the workload by periodically assigning tasks to different cores or moving jobs from one core to another, the temperature difference between the cores will vary.

Temperature variation strongly influences static energy consumption. It is well known that this function is exponential. In the era of high density of the devices on a die, leakage becomes an equally important component as well as dynamic energy, so the effects of temperature should be considered very carefully in the chip design.

2.2.3 Device Ageing

Recent research has shown a high degradation of the devices over time due to Bias Temperature Instability (BTI)[13]. This degradation that can be understood as an increase of the threshold voltage reduces the cell noise margins, increases read access time and increases the number of failures. According to that, transistor ageing should be carefully considered in the sub 20nm design, and adequate mitigating solutions have to be proposed.

Among the theoretical models that try to explain BTI, the most accepted one is the Reaction-Diffusion (RD) model. According to this model, silicon-hydrogen (Si-H) bonds are broken over time at the silicon/oxide interface whenever a negative voltage is applied to the gate of a PMOS transistor. Broken Si-H bonds generate more interface traps. These traps, in turn, capture electrons that flow from source to drain, which eventually leads to an increase in the threshold voltage (V_{TH}) of the transistor. Consequently, the nominal I_d current of the transistor decreases over time which slows down

the circuit.

Similar to NBTI, PBTI (Positive Bias Temperature Instability) affects NMOS transistors whenever a positive voltage is applied to its gate. The causes of PBTI and NBTI are, in essence, the same. Yet, different results can be found in the literature. Initial research mostly showed that effects of NBTI are significantly higher than PBTI [14]. However, recent publications show that the impact of PBTI cannot be neglected especially in the small 10nm FinFET devices [13].

In general, BTI strongly depends on the following factors:

- *Voltage* - The higher the operating voltage, the higher is the BTI degradation. Therefore, lower operating voltages are desired [13]. However at low V_{DD} the susceptibility to V_{TH} variations increases.

- *Temperature* - Research on BTI shows that degradation is higher for higher operating temperatures[14][15].

- *Signal Probability* - Different studies have reported a strong dependence between the amount of BTI degradation and the zero-signal probability [15]. The greater zero signal probability produces higher degradation of PMOS devices due the NBTI.

$$\Delta V_{TH} \propto \frac{qN_{IT}(t)}{C_{OX}} \propto f_{AC}(S_p) \times K_{DC} \times t^n \tag{2.3}$$

In general, the threshold degradation due to BTI can be modelled following Equation 2.3, where N_{IT} is the interfacial trap density; C_{OX} is the oxide capacitance; K_{DC} is a technology-dependent constant, which depends on temperature, V_{DD}, device geometry, etc. f_{AC} function represents the AC dependency of the process [15]. According to the results presented in [15], this feature is approximately linear. In the work that follows, we assume this function to be linear (i.e. for a given technology node, we will consider a linear dependency between the threshold degradation and the signal probability).

2.3 Soft Failures and Errors

A soft error, or single event upset (SEU), is defined as a transient piece of incorrect machine state [16]. Occurrence of soft errors is becoming a critical issue as technology continues to shrink. Due to technology scaling, trends such as smaller supply voltages and reduced capacitance on the nodes cause increased rate of soft errors [17].

Transient faults can be the result of electrical noise, such as crosstalk, or high-energy particle strikes. Soft errors due to energetic particle strikes are typically caused by alpha particles, which can be produced by radioactive contaminants in packaging materials. In the presence of reduced supply voltage, energy that is created by the particle strike can be large enough to flip the state of the node and produce error in the circuit. Besides alpha particle, same effects can be produced by high-energy neutrons from cosmic radiation. While dealing with alpha particles is largely a manufacturing issue, addressing neutron strikes poses a significant problem because adequate shielding is prohibitively expensive[16].

It is very important to design a processor that is going to be able to detect and handle soft error when it happens. These demands are even higher in the environments that are more prone to these effects (e.g. electronics in planes is more susceptible to cosmic radiation) or in the applications where an error could cause tremendous damage (e.g. stock market trading).

2.4 Solutions at Various Levels of Abstractions

2.4.1 Conventional Circuit Techniques to Combat Variability

In order to combat process variability statistical design approach have to be applied in the circuit design. Design space is explored to optimise certain criteria, i.e. energy consumption, reliability, circuit maximal frequency... Majority of those techniques uses transistor size and threshold voltage to tune specific circuit parameters in order to meet target yield. In this section,

we present some of the most famous circuit level techniques that pose state of the art in the cache memory design.

2.4.1.1 Adaptive Threshold and Supply Voltage

Body biasing techniques have been successfully demonstrated for reducing V_{TH} variation. In general, this is the best method to fight variability since it enables the dynamic modification of V_{TH} after chip fabrication that is achieved by adjusting the body-to-source voltage. Forward body biasing (FBB) decreases the threshold voltage V_{TH} of the transistors, increasing maximum frequency and leakage, while reverse body biasing (RBB) has the opposite effect [18, 19, 20, 21]. Tolerance to variations can be increased by utilising both RBB and FBB mutually-exclusively, and this is called Adaptive Body Biasing (ABB) [20, 21, 22, 9].

A similar technique to the classical body biasing can be applied to the SOI FinFETs. It is possible to design SOI FinFET with two gates which give the different design choices. The most common application of this structure is back-gate biasing. This technique involves the following: one gate is used as a control node as in classical devices while the other is biased to the constant voltage. For example, applying a constant negative voltage on one N-type FinFET I_d can be reduced (i.e. V_{TH} is increased).

The linear relationship of dynamic and leakage power with supply voltages can be exploited to control power consumption as in Adaptively Scaling the Supply Voltage (ASV). When compared to ABB, ASV offers less area overhead. However, applying the ASV techniques on caches is very hard since the timing is very critical.

Some methods that use Dynamic Voltage and Frequency Scaling (DVFS) are found in the literature. For instance, in [23], authors present one original solution that dynamically detects timing failure and if the failure is detected voltage is boosted and instructions are repeated for correct computation.

2.4.1.2 Source Biasing

Source terminal of a transistor can be actually used for controlling memory leakage and retention failures [2, 24]. In source biasing technique, when a particular row is accessed, the source line is biased to zero, which increases the drive current and achieves fast read/write operations. When the row is not accessed, the source line voltage is raised to V_{SB}, which substantially reduces both the sub-threshold and gate leakage during the inactive periods. Although increasing the source-bias in an SRAM array reduces the leakage power, the probability of retaining the data at the standby mode decreases (hold failure)[2].

In [24], the effects of source biasing have been investigated on memories under process variations to reduce leakage while maintaining its robustness. The authors propose an adaptive source-biasing (ASB) scheme to minimise leakage while controlling hold failures.

2.4.1.3 Cell Sizing

The length and width of the different transistors of the SRAM bit cell affect the cell failure probability by modifying: the nominal values of access time T_{ACCESS}, the trip point of the inverter V_{TRIP}, the read voltage V_{READ}, the write time T_{WRITE}, the minimum retention voltage V_{DDmin} and the sensitivity of these parameters to V_{TH} variation. The impact of varying the strength of the different transistors on the various cell failure probabilities has been explained in detail in [25].

Although the memory cell characteristics can be improved by increasing the transistor width and length, it effectively contradicts Moore's law (production of transistors with smaller dimensions) in its essence.

2.4.1.4 Assist Techniques

Read and Write assist techniques can be applied to the memory without increasing the cell area which makes them preferable when comparing to the previous ones described in this section. These techniques assume tuning the

bit-cell supply voltage, world-line voltage and bit-line voltage in order to improve read/write margins and V_{DDmin} [26, 27].

Modulating the word-line voltage or pulse width is one such technique. This method decreases the failure rates by reducing the duration or the drive strength of the bit-line on the internal nodes during the read cycle. The length of the word-line pulse can be shortened, which reduces the amount of time that the read stress is applied to cells and thus reduces the failure rates. The read margin can also be boosted by reducing the strength of the access transistor, preventing read disturbance.

The reduction in the SRAM cell supply voltage improves the write margin by reducing the overdrive voltage of the PMOS pull-up transistor. This flips of the high internal node more easily. The virtual supply can be reduced by dividing the array into subarrays and dropping the voltage of only the columns that are being written. A similar scheme utilises the capacitive ratio between the array cell voltage and a dummy voltage line to lower the cell voltage rapidly. Bit-line signals can also be modified in duration or intensity to influence the read and write margins of the cells. For instance, the reduction of the bit-line voltage improves the read margins.

Column sense amplifiers can be implemented in each column to provide full bit-line amplification to improve the read stability. Another approach is a pulsed bit-line (PBL) scheme, where the BL voltage is reduced by 100-300mV just prior to the activation of the WL for 32 nm technology node. This reduces both the cell read-current and the charge sharing between the bit line and the storage node. During the write operation, PBL is applied to the half selected cells (i.e. cells sharing the same WL but belonging to unselected columns). This scheme significantly improves the read failure rate.

2.4.1.5 Memory Cell Modification

Over the years, the 6T cell was the de facto choice for SRAM design. As the need for bigger (and denser) memories is always present, it is of crucial importance that this cell is scaled to the minimum dimensions. However, some constraints have to be satisfied. A detailed explanation of the functionality

of classical 6T SRAM cell can be found in [28].

Failures in 6T-SRAM are characterised into 4 types - *Access Time Failure*, *Read* and *Write Stability Failures* and *Hold Failures*. It is assumed that the primary sources of such failures are the variation due to random dopant fluctuations in the threshold voltages of the six individual transistors that constitute the cell [29].

In order to improve the cell reliability, especially for low power designs, some alternative cells with the higher number of transistors were published in the recent years [30]. For instance, in [31] the asymmetric 7T cell, and in [32, 33] 10T SRAM cells are presented. These cells are designed for low voltage operations. One of the most promising successors of the 6T cell is the 8T SRAM cell [34, 35]. Two more transistors are added (RG1 and RG2) to obtain a separate read path. Because of RG1, the cell "core" is isolated from the output. This significantly improves cell stability. However, the area of the cell is increased, and one more extra line complicates memory routing.

Multiple transistor DRAM cells (Gain Cells) [36, 37, 38] have received a significant attention lately as a potential replacement for the classical 6T SRAM cell. These cells use the capacitance of the device to store the memory state. The charge that flows trough the read line is greater than the charge used for storing the value, so these cells are also called gain cells. A separate read path enables non-destructive reads and reduces the access time when compared to the traditional 1T1C cell. Also, when compared to the 6T SRAM, the different read line isolates the cell core from the read path, and there is no critical read noise margin that should be considered during the read-process.

2.4.2 Micro-Architecture Techniques to Combat Variability

Variation tolerance with low degradation in performance can be achieved by considering cache memory design at a higher level of the design abstraction. In this section, we present some the most famous techniques to combat process variability at the micro-architecture level.

2.4.2.1 Cache Reconfiguration

In [29], Agarwal et. all proposed a process-tolerant cache architecture suitable for high-performance memories that detects and replace faulty cells. It is based on online Built-In Self-Test (BIST) that identifies damaged cells (due to process variation) and a configuration for remapping the faulty block. By using a column MUX, a non-faulty block is accessed when a faulty block is demanded.

In order to map the faulty block to the fault-free block permanently, the tag array width is expanded to two bits (assuming that four blocks share a column multiplexer). These extra bits are set to appropriate values by the configuration controller to ensure that a cache miss occurs when the fault-free block (where the faulty block is mapped) is accessed. This architecture has high fault-tolerant capability compared to the contemporary fault-tolerant techniques, with minimum energy overhead and it does not impose any cache access time penalty. This scheme maps the whole memory address space into the resized cache in such a way that resizing is transparent to the processor. Hence, the processor accesses the resized cache with the same address as in a conventional architecture. By employing this technique, Agarwal et al. [29] achieve high yield under process variation in 45-nm predictive technology.

2.4.2.2 Cache Line Deletion

In [39], a Cache line deletion in suggested when a faulty line is detected (i.e. faulty line is marked and excluded from regular cache line allocation and use). An additional bit is attached to each cache tag [40] to determine the faulty cache. Any cache line with the availability bit turned off is defective and is not used. A similar strategy has been employed in IBMs Power4 processor, which can delete up to two cache lines in each L3 cache slice [40].

2.4.2.3 Error Detection and Error Correction Codes

In this paper we

Error detection is the most important aspect of fault tolerance because a processor has to be aware of a problem before it can tolerate it. Error De-

tection Code (EDC) has been incorporated into most commercial computers [41, 42]. Moreover, the most of the processors incorporate Error Correction Code (ECC) in every level of cache hierarchy (e.g. in [43, 44].

Most caches are write-back, in which modified data is not immediately propagated to lower levels of the memory hierarchy. Since modified data has no back-up copies in other levels of the memory hierarchy, write-back caches are especially vulnerable to soft errors[41]. Because of that the most of write-back caches comprise Single Error Correction Double Error Detection (SECDED) code. However using ECC in the cache implementation tradeoffs are made since ECC implementation increases cache area and reduces cache speed. One interesting solution was shown in [45]. The paper presents error coding for spatially correlated errors in two-dimensional coding scheme.

2.4.2.4 Micro-Architecture Techniques for Reducing Cache Energy

Caches are the highest consumer of the energy (especially leakage energy) because they occupy the greatest part of a chip area. Among all methods proposed to reduce the cache leakage, here we mention two, the most referenced since many other works that have appeared latter have extended these ideas at a certain level.

In [46], authors suggest moving the particular lines of the cache to the low power *Drowsy State*. They present different policies for transferring unlikely used lines into the Drowsy state. According to their work, they achieve a significant energy reduction since the great part of the cache lines (more than 80%) is unused or very unlikely used in standard cache behaviour.

Another work presented in [47], is based on the idea that a line, when it is pulled in the cache, is more likely to be used to some predefined time. Decay exploits this idea by invalidating and "turning off" cache lines after some predefined period of time.

Besides for reducing leakage energy, ideas that extend Drowsy and Decay caches fundamentals can be used for reducing the refreshing energy in the dynamic memories. This approach will be presented in Chapter 6.

And now, for something completely different.

Monty Python's Flying Circus

3

FinFET Technology

3.1 Introduction

Introduction of the multi-gate FET technology has prolonged Moore's law by solving the problem of Threshold Variability caused by Random Dopant Fluctuations. Better channel control that is achieved by fabricating the transistor gate from the more than one side is the key.

Among all Multi-Gate devices, Tri-Gate FinFETs emerged as the most promising successor of the traditional bulk technology. The primary reason for this is the development process of these devices that is slightly more complicated than of the traditional planar devices, yet much simpler than for some other multi-gate structures (e.g. Surrounding Gate structures).

In this chapter, we present the fundamental facts about Tri-Gate SOI FinFET technology. We show detailed characteristics of BSIM-CMG model card developed by the University of Glasgow, Device Modelling Group that we used in our research. The main contribution of this is work is the proposal how the variability of FinFET devices can be incorporated in an HSPICE model card for time efficient circuit simulation when classical Monte Carlo methods are used. We also presented how the effects of the back-gate biasing

(that can be applied in Independent Gate FinFETs) can be simulated with the Tri-Gate model card that was available for this research.

This work has been published in the Conference "Mixed Design of Integrated Circuits and Systems" (MIXDES 2012) [48] and journal "Transactions on Electron Devices (TED 2013)" [49].

3.2 Characteristics of FinFET Technology

3.2.1 Tri-Gate FinFETs

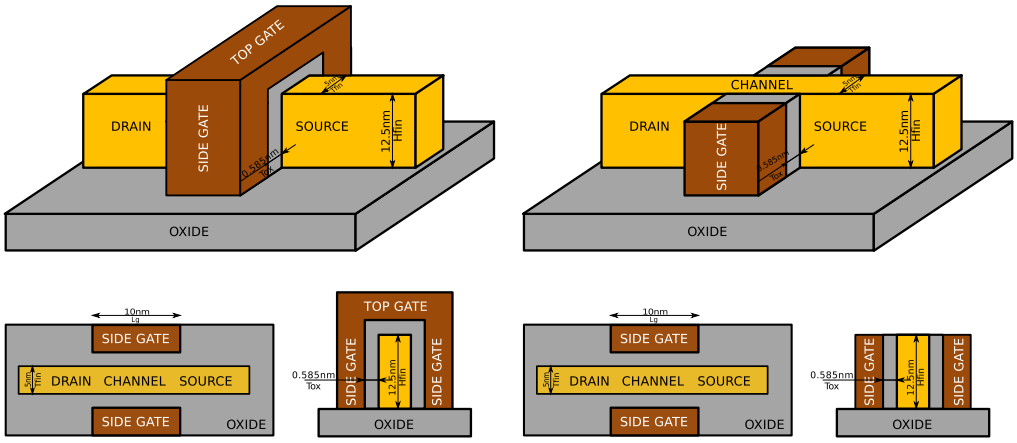
The main characteristic of Tri-Gate FinFET devices (Figure 3.1a) is that the gate wraps the channel from 3 sides. This increases the effective channel width that gives better control of the short channel effects (SCE), and a high dopant concentration is not necessary anymore - as it was the case with traditional bulk MOSFETs [50, 4]. As a consequence, two advances can be noticed: (*i*) the variability caused by random discrete dopants becomes insignificant; (*ii*) the leakage current is lower comparing to the classical bulk technology [6].

The major difference between the FinFETs and the classical planar devices is the quantisation in the FinFETs channel width that can be calculated according to the Equation 3.1.

$$W_{eff} = N_{fin}(2H_{fin} + T_{fin}) \tag{3.1}$$

where N_{fin} is the number of fins of the transistor, H_{fin} fin height and T_{fin} fin thickness.

Our model is designed for a 10nm channel length (L); fin height (H_{fin}) and fin thickness (T_{fin}) are 12.5nm and 5nm respectively. Effective oxide thickness (EOT) is 0.585nm. Detailed transistor parameters are presented in Table 3.1.



(a) Tri-Gate Fin-FET (b) IG-Fin-FET

Figure 3.1: 3D Structure of SOI Fin-FETs

Table 3.1: The FinFET parameters

Node (nm)	**11**
L (nm)	10
EOT (nm)	0.585
T_{fin} (nm)	5
H_{fin} (nm)	12.5
N_{SD} (cm^{-3})	3e20
N_{CH} (cm^{-3})	1e15
V_{dd} (V)	0.8
I_{off} $(nA/\mu m)$	97
I_{dsat} $(\mu A/\mu m)$	1958
SS (mv/dec)	73.2
DILB (mV/V)	47

3.2.2 Independent Gate (IG) FinFET

Specifically, the top gate of the channel can be etched and by that, two independent gates can be formed as in the Figure 3.1b [4].

The most common application of this structure is called back-gate biasing that involves the following. One gate is used as a control node as in classical devices and the other is biased to the constant voltage to control transistor strength (e.g. applying a negative voltage on one N-type FinFET gate, I_d

is reduced). This technique is similar to the body biasing in classical bulk technologies [19, 18].

For the simulation of this technique by BSIM-CMG model card, we used the following approach. A voltage generator is connected in series with the FinFET control gate to emulate the effects of back-gate bias. In general, the voltage value of this generator is dependent on the potential difference between source and gate (V_{GS}) that the best I_D curve fitting can be achieved. For simplicity, we assume that this voltage is constant (independent of V_{GS}). This should not induce a big error in simulations if it depends on the active part of the $I_D V_{GS}$ curve. For instance, this method is used in Chapter 4 where we simulate the stability (RSNM, WLWM) of the 6T SRAM cell.

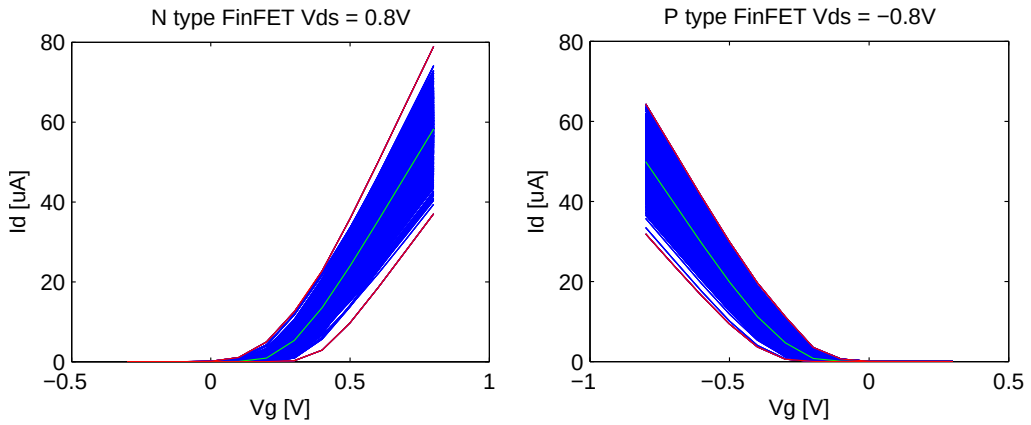
3.3 FinFET variability

Although FinFETs have alleviated the problem of the threshold variability caused by RDF, some sources of variability still remain. Variation in circuit parameters resulting from the effects of Fin Edge Roughness (FER) and Gate Edge Roughness (GER) still pose significant challenge for these devices [7, 6], and its effects are going to worsen as the channel length approaches 10nm [6].

Numerical simulations based on finite-elements methods or TCAD (Technology CAD) tools are useful for technology evaluation. TCAD simulations can be combined with Monte Carlo simulations to predict the impact of the device variation on the circuit performance. However, these simulations are very time-consuming and applying them to a circuit with a large number of transistors is impossible [50, 11]. Because of that analytical models are developed. Analytical model comprise set of complex mathematical equations that describes physical process of the device. In general data obtained by the TCAD simulations (or physical measurements of real devices) are fitted by set of mathematical equations by carefully setting appropriate parameters.

BSIM-CMG ([11]) is the new generation surface-potential-based multi-gate devices model, which is available in HSPICE [51]. In this section, we propose a procedure to obtain variability characteristics from TCAD simu-

lations in order to incorporate them in an HSPICE simulation.



(a) N-Type (b) P-Type

Figure 3.2: Tri-Gate 10nm SOI FinFET Transfer Functions

Figure 3.2 presents the transfer characteristics of the state-of-the-art 10nm SOI FinFET devices featuring metal gate and high-k dielectric with the default temperature of 30°C. Device parameters are shown in the Table 3.1. Plots have been obtained by TCAD simulations when the variability of RDD, FER, GER, metal gate granularity (MGG) and interface trapped charge (ITC) are incorporated. Variability amounts are extracted from the ITRS report. The device is designed to meet the requirements set by ITRS roadmap for high-performance applications. The GSS "atomistic" simulator GARAND [52] is used to investigate the statistical variability and reliability. For both FinFET types, 1000 instances are analysed to minimise statistical error (approximately 95% confidence for 3σ). Nominal supply voltage is $V_{DS} = 0.8V$. The random dopants are introduced based on the nominal local doping concentration. LER is obtained from Fourier synthesis with Gaussian autocorrelation, parameterised by correlation length (30nm) and root mean square (RMS) varied in the simulations [53]. TiN gate metal grains are assumed to have two different possible crystalline orientations with different work-functions spanning 0.2V and having 40% and 60% probability of occurrence [54, 55]. To introduce a random grain pattern into the gate of each simulated device, a similar procedure to the one, for investigating polysilicon

and high-*kappa* granularity (described in detail in [56]), is applied. The detailed simulation procedure, technology assumptions and individual or combined effects of variability source on the device characteristics can be found in [6].

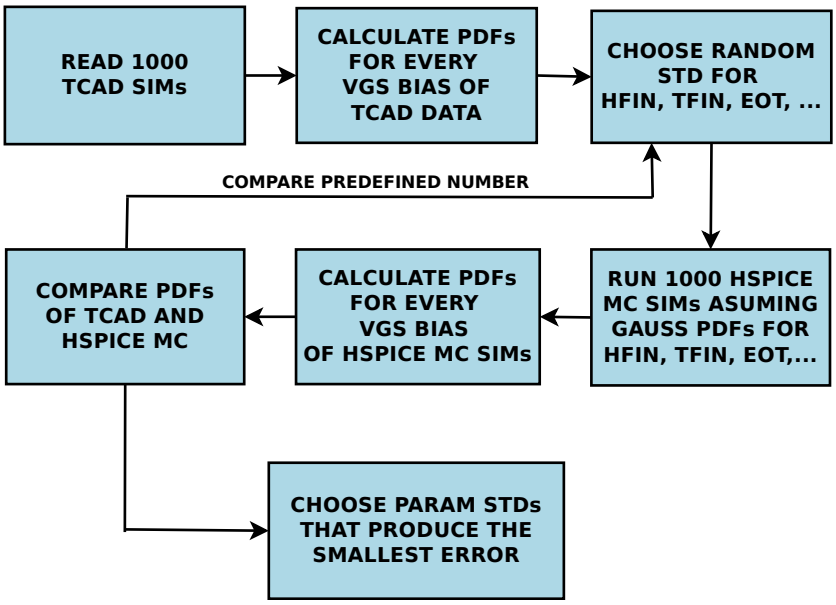


Figure 3.3: FinFET variability calibration procedure

The most common procedure for circuit variability simulation using the HSPICE is to do Monte Carlo simulation when some of the model parameters are Gaussian random variables with mean equal to the nominal value [3, 57].

Variability simulation of traditional bulk structures is mostly based on the variation of V_{TH}. Although in [6] authors show that individual sources of process variation produce approximately an asymmetrical distribution of V_{TH}, for simplicity reasons, we decided to model process variability as a Gaussian distribution of few parameters from HSPCE model card. As we will see, in short, the simulation error is less than 2% under this approach.

Since BSIM-CMG is surface-potential-based model, there is no V_{TH} parameter and an alternative for process variability simulation has to be found. Our proposal is to consider process variability as the variation in H_{fin} (fin height) and T_{fin} (fin thickness), EOT (oxide thickness), L (channel length). We assume a normal distribution of these random variables with the mean

value equal to the nominal value of the device parameter. For searching optimal standard deviations we used simple heuristic search. We were finding standard deviation of the device parameters from the specific ranges that we obtained by the initial characterisation of the device transfer characteristics. When this initial search has been done we approached detailed and more accurate search. The procedure to determine the standard deviations of these random variables that would approximate process variability with the highest accuracy is illustrated in Figure 3.3.

The first step is to load 1000 TCAD transfer characteristics data (1000 $I_D V_{GS}$ plots obtained by the TCAD variability simulation of the FinFET devices). Next, calculate I_D frequencies for every value of gate-source bias. Next, we pick up a random value of standard deviation for different parameters in HSPICE model card (H_{fin}, T_{fin}, L and EOT). Then, perform HSPICE DC simulation with 1000 Monte Carlo samples and compare frequency distributions of I_D for every value of gate-source bias. This procedure is repeated iteratively until the difference (or similarity) between the outputs is below a certain threshold. For the measure of similarity (i.e. error) between distribution functions, we choose a number of values that fall in 3σ and 5σ range of the I_D distribution function that is obtained by TCAD simulation.

After the iterative process, this procedure achieves an error (as defined in the previous paragraph) of less than 5% for 3σ and less than 2% for 5σ, when T_{fin} and H_{fin} parameters are considered as Gaussian random values with relative standard deviation of 16% and 12%. Simulation has shown that minimal error is reached when just T_{fin} and H_{fin} parameters are considered as Gaussian variables. Because of that, we shall consider L and EOT constant in further analysis. Here it should be noticed that process variability could be regarded as the variation of some other parameter from the BSIM-CMG model card (e.g. variation of gate work-function-PHIG). However, the previously developed method that considers process variability just as variation of T_{fin} and H_{fin} parameters, approximate I_D distribution functions with very small error (less than 2% for specific error measure as previously claimed). For that reason, our variability simulations use just these two parameters.

In the case of independent back-gate biased FinFET, we assumed the

same variability metrics as for Tri-Gate FinFET, which is realistic considering that the same device dimensions are used.

3.4 Conclusion

We have shown an original method to incorporate process variability of the 10nm FinFETs, previously obtained by TCAD simulations, in an HSPICE BSIM-CMG card for time-efficient simulation without compromising results. BSIM-CMG became standard when simulating multi-gate devices, and because of its complexity and distinction from threshold based MOSFET models, it is of crucial importance to find an alternative method to incorporate process variability in the simulation. In this chapter, we show that by varying just two parameters H_{fin} and T_{fin} very accurate results can be achieved when classical Monte Carlo simulations are used. Comparing to the results obtained by TCAD simulations this method shows error less than 2% in 5σ ranges. Moreover, we have shown one simple solution that can be used for simple simulation of back-gate biasing for the Independent Gate FinFETs, that can be used to give certain insight about technology when only Tri-Gate models are available.

This used to be a nice neighbourhood before the old ladies started moving in.

Monty Python, Gangs of Old Ladies

4

FinFET SRAM Cells

4.1 Introduction

Novel multi-core processor architectures demand more on-chip caches for efficient sharing information across parallel processing units. These memories are traditionally designed on SRAMs. Since they occupy the greatest part of the chip area, it is of crucial importance that the SRAM cell, as their primary building unit, is scaled to the minimal dimensions. Parameter fluctuations have the greatest influence on circuits that rely on perfectly matched transistor pairs [8]. If the small dimensions of SRAM cells are added to the previous statement, it becomes evident that process variations have the greatest influence on them. However, some constraints have to be satisfied and trade offs has to be searched.

In this chapter, we characterise the 6T and the 8T SRAM cells designed in the 10nm SOI FinFETs technology. Ideally, the cells are scaled to the minimal dimensions that allow sufficient density while keeping performance (i.e. speed and power) within the design goals. Starting from the minimum sized SRAMs, we analyse their performance when exposed to the different sources of the process variability: FER, GER, MGG and environmental variations

(i.e. supply voltage and temperature). Since there is still little information regarding the actual FinFET fabrication process, in our analysis we considered only intra-die variability sources. Also, since FER, GER and MGG are major source of the process variability in these devices, and since the dimensions of the cell that we simulate are very small, it is very reasonable to assume that these process variations are random.

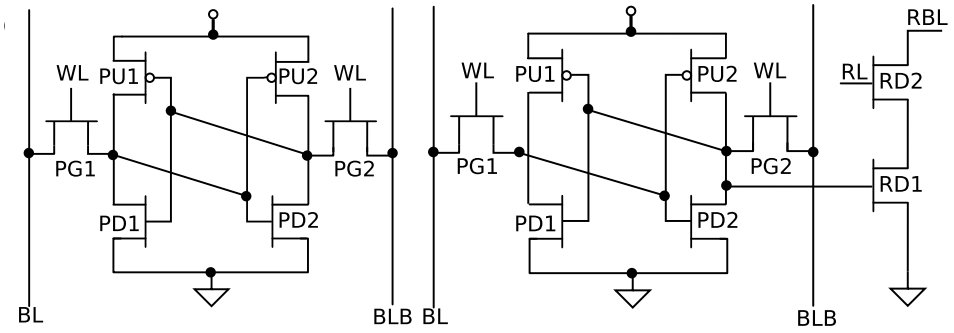
Besides that, we analysed a technique for World Line Write Margin (WLWM) and Read Static Noise Margin (RSNM) enhancement by lowering the strength of the pull up (PU) and the pass gate (PG) transistors. This can be achieved using a reverse back-gate biasing technique when the independent gate FinFETs are employed in an implementation. Using the back-gate biasing is a very attractive method to fight variability since it can be applied dynamically after chip fabrication.

The main contribution of this work is the characterisation of the 6T and the 8T SRAM cells for the 10nm SOI Tri-Gate FinFET technology under process and environmental variations. While we make a thorough analysis of the behaviour of minimum sized 6T and 8T cells, we also provide a study of different design points (i.e. cell sizes). We show how the RSNM and WLWM of the 6T SRAM cell are improved when the back-gate biasing is applied in the cell design. Finally, we also compare to the previously published work on the SRAM cells in higher technology nodes to provide an insight into the scaling trends of the FinFET SRAMs.

This work has been published in the Conference Mixed Design of Integrated Circuits and Systems" (MIXDES 2012) [48] and journal "Transactions on Electron Devices (TED 2013)" [49].

4.2 FinFET SRAM Cells Characterisation

For proper dimensioning of the cell, two fundamental constraints have to be satisfied. The PU transistors have to be weaker than the PG transistor (*write-ability*), and the PD transistor has to be stronger than the PG (*readability*). A detailed explanation of the functionality of the classical 6T SRAM cell can be found in [28].



(a) Classical 6T SRAM cell (b) 8T SRAM cell

Figure 4.1: SRAM cells

Read static noise margin (RSNM) is defined as the minimal noise voltage that can flip the cell state during the reading process. RSNM is calculated as the maximum square that can fit in the butterfly curve. The butterfly curve is formed by plotting the voltage transfer characteristics of the two inverters in an SRAM cell when both bit lines (BL and BLB) and the word line (WL) are biased to V_{DD}. If the two squares inside the butterfly curve do not have equal side lengths, the RSNM is the side length of the smaller one [11].

Word-line write margin (WLWM) characterises the cell's write stability. A write operation in SRAM is typically carried out at a WL voltage of V_{DD}. In some cases, the WL voltage can be lowered during the cell access in order to improve cell RSNM or reduce dynamic power consumption. WLWM is the maximal value by which the WL voltage can be reduced below V_{DD} that still allows a successful write [11].

The operation of a 6T FinFET SRAM cell is the same as the conventional 6T SRAM planar circuit, and the same constraints have to be satisfied. However, the dimensions defined by bulk devices cannot be directly applied to FinFET technology due to the quantisation of the FinFETs width [4, 58]. The smallest cell has one fin in the PU and the PG transistors. Read stability can be improved by increasing the number of fins in the PD transistors [4, 58]. In this chapter, we evaluate the performance of the 6T SRAM cell that has 2 or 3 fins in the PD transistor.

In order to improve further RSNM, especially for low power designs, some

alternative cell designs with a higher number of transistors were published in the recent years [30]. In this work, we also present results for the 8T SRAM cell (Figure 4.1b) which has shown to be the most promising successor of the classical 6T cell. Two more transistors are added (RG1 and RG2) to obtain a separate read path. Because of the RG1, the cell "core" is isolated from the output. This significantly improves the read stability and only one fin in the PD transistor is needed to achieve high RSNM values. In our analysis, we consider an 8T SRAM cell that has only one fin in the PD transistors.

Besides the already mentioned RSNM and WLWM constraints, in this work we present results for two more metrics.

Leakage (the term leakage is used as a synonym for the static power) is the cell power consumption when it is not accessed (when the WL is set to 0). Inactive power consumption is not 0 because of the leakage currents. As the technology has been shrunk beyond 32nm, this component has become a limiting factor in the design process.

Read Access Time (RAT) is defined as the time that elapses from the moment when the cell is activated until the moment when the Bit-Line (BL) voltage reaches some predefined value. Similar to the RAT, we can define *Write Access Time (WAT)* but every cell is always more sensitive to the reading process (RAT is always greater than WAT). According to this, in order to compare the cells speeds in this chapter we only present results for the RAT.

4.3 Related Work

Because the SRAM cell is the most sensitive component in the design process, its characterisation is of crucial importance. Additionally, measuring the stability and the leakage power of the cell gives a good estimation of the technology, especially under the effects of process variation. In this section, some significant work related to the silicon technology evaluation and the SRAM cell characterisation is presented.

In [59, 4, 58], authors characterised FinFET SRAM cells for 32nm technology when Tri-Gate (TG) and Independent Gate (IG) FinFETs are used.

They show how RSNM can be increased up to 220mV when three fins are used in the PD transistor. Mainly, they presented one realisation of an IG FinFET SRAM cell that can improve RSNM up to 240mV (for power supply of 0.8V) with 17% of area and 52% power consumption reduction at the same time. RSNM improvement, by reducing the PG strength, was presented in [60], too. Authors used the 32nm models for their work and for the 1V power supply they show that the RSNM can be increased up to 325mV when IG FinFETs are used in the PG transistors for dynamic V_{TH} control comparing to 233mV when that is not the case.

Similar work has been done further for smaller technology nodes so the following papers can be found. In [61, 62] RSNM and WLWM of different 6T SRAM cell types, are evaluated for 300mV and 500mV respectively for 25nm FinFETs under process and environmental variations. Slightly smaller devices are presented in the works of Carlson et al.[63] and Shin et al., [64] for the 22nm technology node. According to these papers, RSNM can be increased between 270-300mV when IG FinFETs are used. In [65] 18nm TG FinFETs that incorporate, high-κ dialectric for surface isolation are used for the 6T cell design. Results have shown up to 160mV RSNM, 300mV WLWM with leakage less than 10nW per cell. SRAM cells characterised for 13nm devices are presented in [66, 67]. Results show that a leakage of 0.7nW and RSNM of 338mV can be achieved for the 8T SRAM cell, and similar results are obtained for one particular IG 6T SRAM cell. All these cells are designed to work on power supply 1V-1.1V.

As far as we know, the smallest designed FinFET SRAM cells are presented in [68]. Gupta et al. developed different 6T SRAM cells with 10.8nm TG and IG FinFETs for 0.7V supply. They show that for the particular case of IG 6T SRAM cell, RSNM can be increased up to 250mV (comparing to 150mV for classical TG cell) but at the cost of WLWM (280mV for TG compared to 150mV for IG). The estimated leakage current is 2.5nA.

The main contribution of this work is the evaluation of the technology that is available to us for which we believe that is going to be one of the mainly used in IC design in the near future. We used Tri-Gate SOI FinFET models for 10nm devices which are the smallest currently available. Doing

this analysis is of crucial importance because the most cache memories in current processors are implemented in SRAM technology. Also, this study will serve as a baseline for the alternative solutions that are going to be presented in the following chapters of this document.

4.4 Simulation Results

We performed the simulations for different supply voltages and temperatures for the SRAM circuits exposed to process variability. For every case (of supply voltage and temperature) we simulated 1000 Monte Carlo instances are simulated to achieve high confidence. Parameters T_{fin} and H_{fin} are considered Gaussian random variables with relative standard deviation of 16% and 12% respectively (Section 3.3). Because LER is the leading cause of this process variability, these random variables have been considered entirely independent from fin to fin [11].

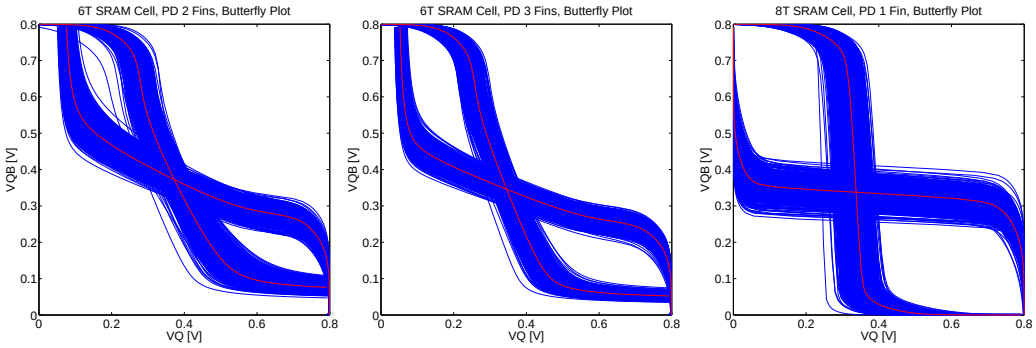
For the presentation of the results bar plots are used. The mean value of the particular metric of 1000 simulated samples is presented. Also, every plot holds an error range $(\mu - 1.96\sigma, \mu + 1.96\sigma)$ which corresponds to the 95% confidence interval.

4.4.1 Read Static Noise Margin

Butterfly plots for the different cell type and size are presented in Figure 4.2. Red lines on every figure are ideal butterfly characteristics (cell without variability).

Read Static Noise Margin (RSNM) dependence on power supply and temperature is presented in Figures 4.3 and 4.4. It can be noticed that median value of RSNM can be increased by 25mV(15%) when used 6T cell with 3 fins in the PD or by 84% by the 8T cell, compared to the results with a 6T cell with 2 fins in the PD. Also, RSNM shows a slight dependency of temperature (15mV for 30-110°C).

When compared with previous work, it can be seen that the median value of RSNM (that lays in the range from 170mV for 6T SRAM cell with 2 fins to



(a) 6T, PD 2 Fins (b) 6T, PD 3 Fins (c) 8T, PD 1 Fin

Figure 4.2: SRAM cells butterfly plot

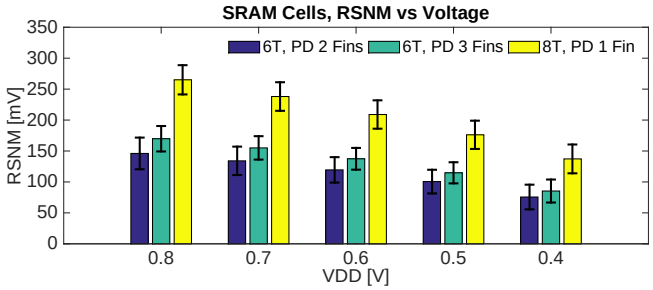


Figure 4.3: SRAM cells Read Static Noise Margin (RSNM) vs. Vdd

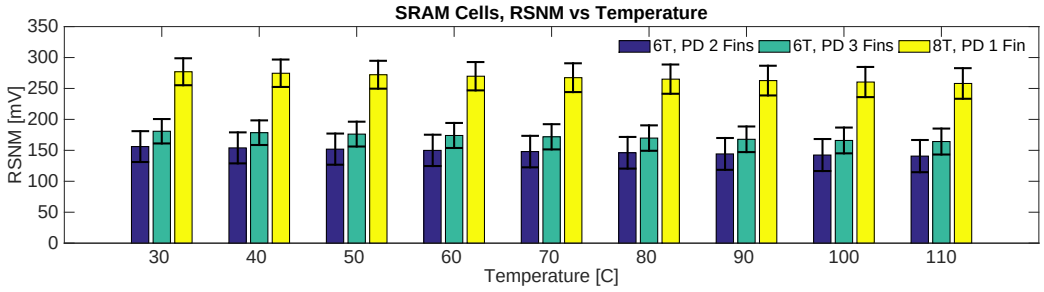


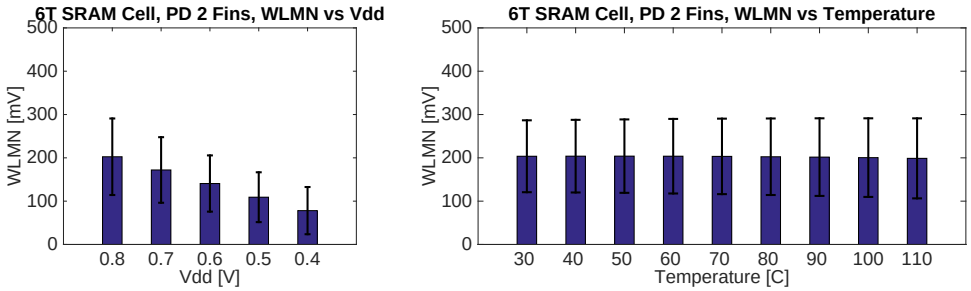
Figure 4.4: SRAM cells Read Static Noise Margin (RSNM) vs. temperature

the 305mV for 8T SRAM cell) is similar to the results presented in the related work. However, most of those calculations are done for a supply voltage of 1V for technologies with higher transistor channel lengths [58, 61, 62, 65, 66, 67]. According to that, we can conclude that RSNM is not significantly degraded in smaller technologies and even with a reasonable supply voltage scaling to

0.8V.

According to [69], in order to ensure SRAM stability $\mu - 6\sigma$ of RSNM should exceed 4% of V_{DD}. This criterion has already been used in the initial evaluation of 6T FinFET SRAM cell presented [70], so we mention it here, too. According to this norm, a 6T cell that has 2 fins in the PD transistor is not enough for $V_{DD} = 0.4V$, while 6T with 3 fins in the PD passes the test for $V_{DD} = 0.4V$, and 8T even for $V_{DD} = 0.3V$. However, as we will see in the next section, scaling the supply voltage to these extremes is not possible due to the World Line Write Margin (WLNM) limitations.

4.4.2 World Line Write Margin



(a) WLNM vs. Vdd (b) WLNM vs. Temperature

Figure 4.5: 6T SRAM Cell World Line Noise Margin (WLNM) vs. Vdd and temperature

WLWM simulation is obtained by setting the BL to 0 and BLB to V_{DD}. The WL voltage (V_{WL}) is swept from 0 to V_{DD} and the value when the cell flips its state is measured. The difference between V_{DD} and the measured value is WLWM.

WLWM dependence on V_{DD} and temperature is presented in the Figure 4.5. The results show very similar behaviour for the 6T and the 8T SRAMs, a subtle dependency on fin number in PD transistor and chip temperature.

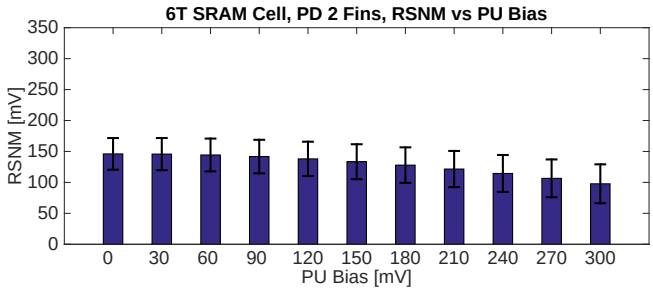
Comparing the WLWM results with previous works [61, 62, 65], it can be seen that the median value of WLWM is approximately 200-300mV lower. However, these measures are made for 1V supply voltage. Given the observed

linear relation between WLWM and supply voltage, we can conclude that technology scaling does not affect directly WLWM, but supply voltage scaling does. For a supply voltage of $V_{DD} = 0.4V$, a couple of samples fail to complete the writing process (from 1000 simulated), which is unacceptable.

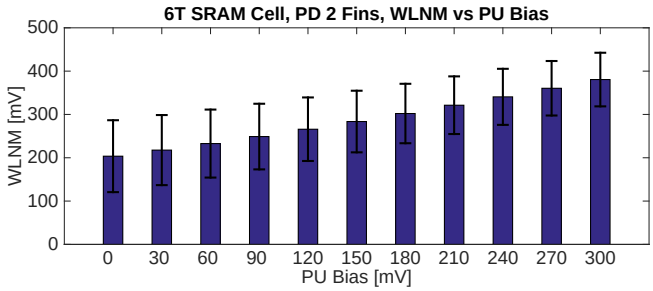
4.4.3 Increasing the Cell Stability by Back Gate Biasing

In this section, we show how the RSNM and WLNM of the cell can be increased when back-gate biasing is applied to the PG and the PU transistors for IG FinFET technology. The cell is designed for an 0.8V power supply. Plots are made for different cases of the PU and the PG back gate biased voltage. We analysed the 6T cell with two fins in the PD as it can be considered optimal regarding stability-area trade off [4].

4.4.3.1 Increasing WLMN by Applying Positive Bias on Back Gate of the PU Transistor



(a) RSNM vs. PU Bias



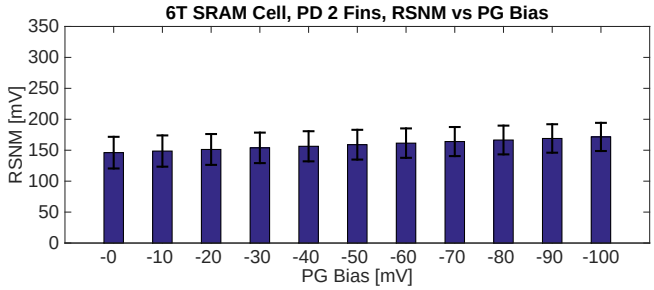
(b) WLNM vs. PU Bias

Figure 4.6: 6T SRAM cell stability vs. PU bias

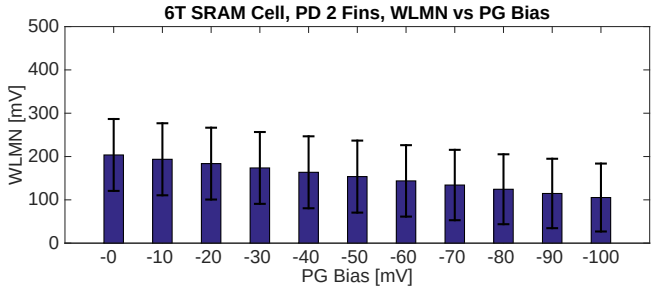
Figures 4.6 show how the WLWM and RSNM change when the strength of the PU transistor is reduced as a reverse back gate bias is applied on it. It can be seen that the WLNM can be increased by almost 190mV while at the same time RSNM is lowered by 50mV.

4.4.3.2 Increasing RSNM by Applying a Negative Bias on the Back Gate of PG Transistor

Figures 4.7 show the RSNM and WLMN when a reverse back gate bias is applied to the PG transistor. It can be seen that the RSNM can be increased some 25mV but, in that case, the mean value of WLMN drops to 100mV.



(a) RSNM vs. PG Bias



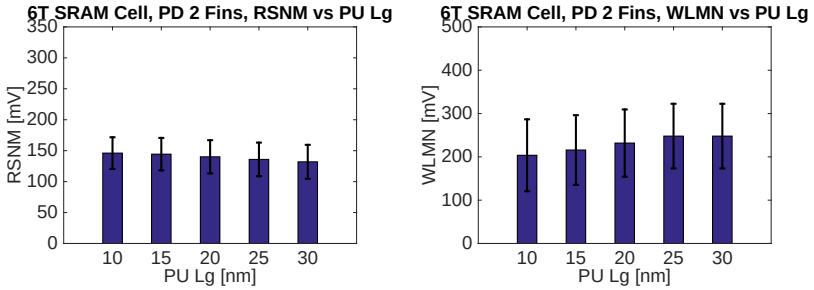
(b) WLMN vs. PG Bias

Figure 4.7: 6T SRAM Cell Stability vs. PG Bias

4.4.3.3 Increasing WLMN by Increasing the Gate Length of the PU Transistor

An increase of the WLNM can be achieved by increasing the channel length of the PU transistor. By increasing the channel length of the PU transistor, its

strength is reduced. The effects of this approach are the same as the effects of applying a positive bias on the back gate of PU transistor. the positive side of this technique, when compared to the PG and the PU biasing is that it can be applied on both, Tri-Gate or Independent-Gate transistors. However, its application is limited by the design process (i.e. once set the transistor dimensions are fixed and further adaptation is not possible).



(a) RSNM vs. PU Lg (b) WLMN vs. PU Lg

Figure 4.8: 6T SRAM cell stability vs. PU channel length

4.4.4 Read Access Time

In this section, we present the results for the cell Read Access Time (RAT). In order to make this type of measurements, we performed a transient analysis using the HSPICE simulator. In our simulation, we observed one memory column when the BL and BLR are connected to a capacitance of 5.12fF which should be capacity of a column with 64 SRAM cells according to the ITRS.

We measured RAT as the time from the moment when WL line is activated (when the voltage on the WL node reaches $V_{DD}/2$ to the moment when voltage on the BL reaches $5/8V_{DD}$ (approximately $V_{DD} - V_{TH}$). Figures 4.9 and 4.10 show that RAT has a minuscule dependency to temperature. However, a significant increase in cell access time when the supply voltage is reduced can be observed (the strength of the PD and the PG transistors reduce, and time to discharge the BLR capacitance is longer). Also, the cell RAT is dependent on the number of fins in the PD transistor for 6T cells, so the cell with three fins in the PD is the fastest. On the other hand, the 8T cell is the slowest since the read out path has only one fin transistors in

read path. RAT of the 8T cell can be reduced by inserting an additional fin in the RD1 transistor, but this has the area penalty. The RAT of the 8T cell with two fins in the RD1 transistor is almost the same as the 6T with two fins in the PD.

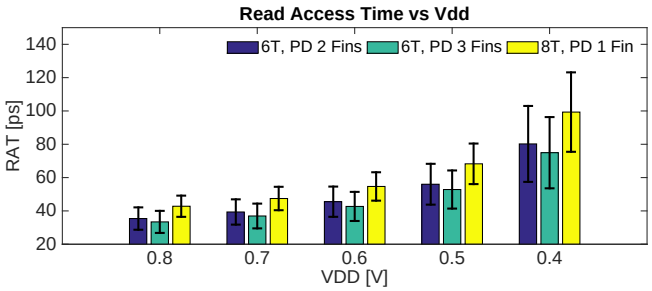


Figure 4.9: Read Access Time vs. Vdd

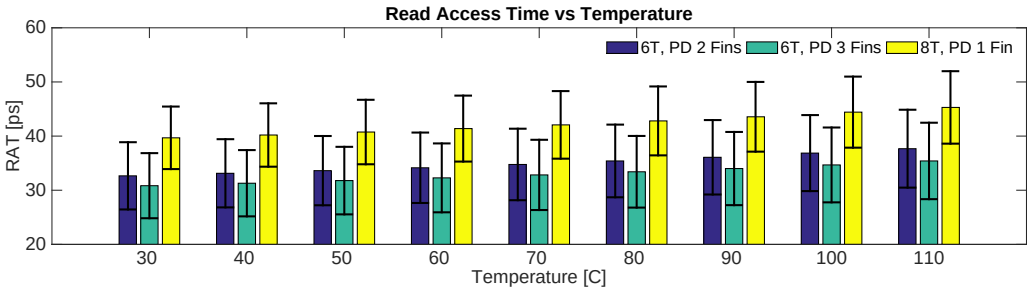


Figure 4.10: Read Access Time vs. Temperature

4.4.5 Static Power Consumption

In this section, we present the results for the leakage evaluation of the one memory block.

In order to estimate the leakage of the entire memory block, we adopted the following methodology. We estimated the leakage of one cell when it is not accessed. When the WL is set to 0V, and every BL and BLR have V_{DD} applied on their nodes. We measure the static power consumption of one cell. Also, 1000 Monte Carlo samples are simulated in order to estimate the mean leakage of one cell.

Since the variation of the memory block is assumed to be entirely random, the mean value and standard deviation of the leakage can be calculated

according to the Equations 4.1 and 4.2 where N is the number of the cells in the block. For a large N, it can be assumed $P_\sigma(Mem)/P_\mu(Mem) \to 0$

$$P_\mu(Mem) = N \times P_\mu(Cell) \tag{4.1}$$

$$P_\sigma(Mem) = \sqrt{N} \times P_\sigma(Cell) \tag{4.2}$$

Figures 4.11 and 4.12 present the results of the static power consumption of the 4KB memory block.

It can be noticed that the leakage of the 8T cell is approximately same as the 6T cell with 2 fins in the PD transistor. The leakage of the 6T cell with 3 fins in the PD is the largest (22% larger than the 6T with 2 fins in the PD).

When compared to the leakage results from related papers [67, 68, 66], we noticed an increase in static power with the technology node.

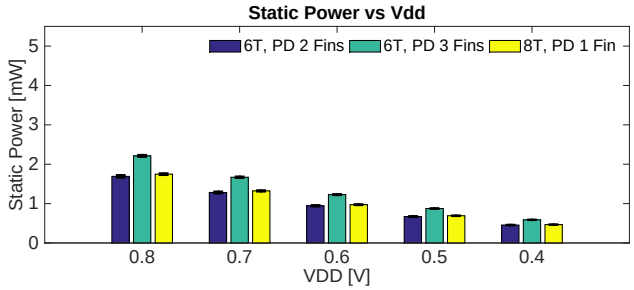


Figure 4.11: Static power of a 4KB SRAM block vs. Vdd

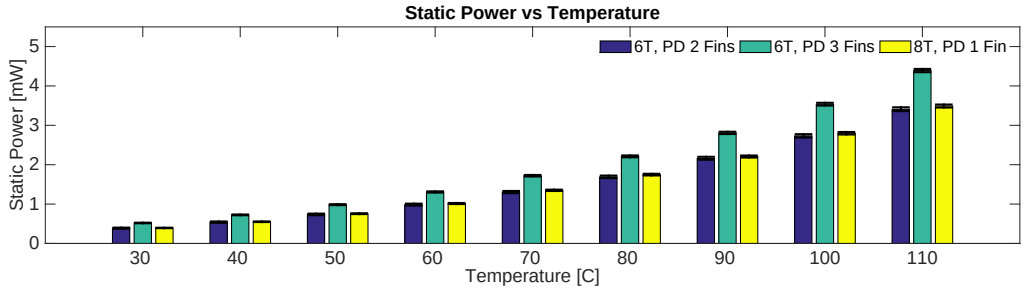
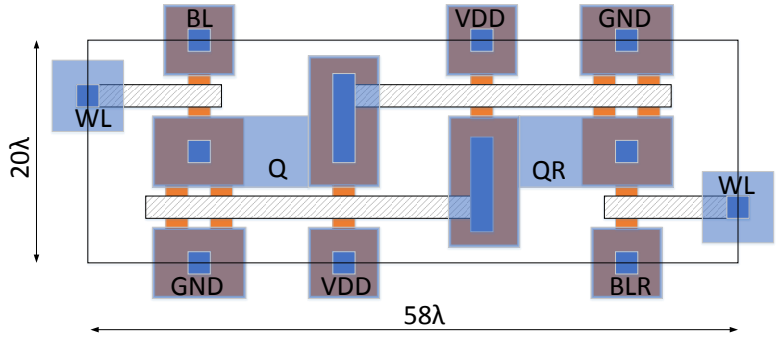


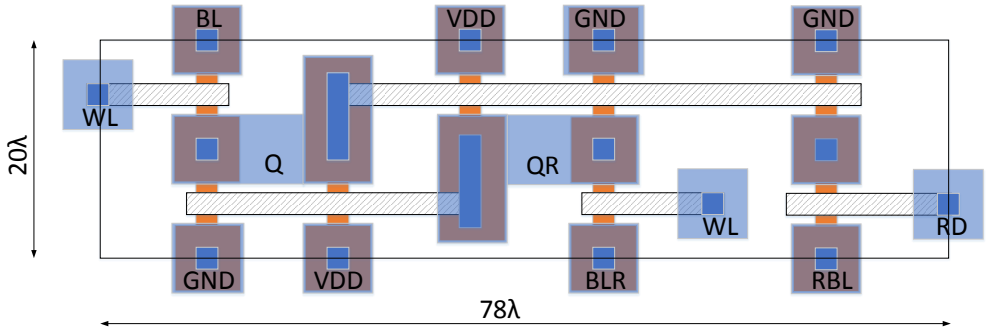
Figure 4.12: Static power of a 4KB SRAM block vs. Temperature

4.4.6 Cell Layout Analysis

In this section, we present a layout analysis. We sketched two cells: the 6T cell with two fins in the PD transistor and the 8T cell. For the layout design, we followed the general rules that are presented in [28] and for FinFET specifics we also used [71]. For determining the fin pitch between transistors that have more than one fin (the PD in the 6T cell) we used ITRS predictions [1]. The layouts are presented in the Figure 4.13.



(a) Classical 6T



(b) 8T

Figure 4.13: Layouts of SRAM cells

Figure 4.13 shows that the area of the 8T cell is significantly larger than the area of the 6T cell (approximately 35%). Although the layout of the 6T cell with 3 fins in the PD transistor is not sketched, its layout is very similar to the 6T with 2 fins in PD. Due to the bigger PD transistor, the overall area of the cell increases approximately 14%.

It is good to mention here that these layouts are drawn assuming SOI devices. If the bulk devices had been used instead, the area would be slightly Bigger as PMOS devices would need a n-well. This increases the width of the cells by 8? in total according to [28]

4.5 Conclusion

In this chapter, we presented the results of the characterisation of the FinFET SRAM cells designed for Tri-Gate FinFET technology. We evaluated the stability, leakage and access time of two types of SRAM cells when they are scaled to the minimal dimension of the channel length.

Our analysis of the SRAMs confirms the good stability of these devices when they are exposed to the effects of process and temperature variations. Nevertheless, the RSNM of the 6T SRAM cell can be improved by increasing the number of fins in the pull-down transistors. However, this increases leakage and area. Specially, the RSNM of the 6T SRAM cells can be improved up to 25% without compromising write-ability when back-gate biasing is applied to the PG transistor for IG FinFETs, and the WLNM can be improved 90% by applying reverse bias on the PU transistor. By combining these techniques we achieved better stability. According to the simulations, the application of the PU bias is more beneficial since it increases the WLNM that is more critical for these devices than the RSNM for these devices.

On the other hand, the 8T cells are more robust to the process variations, and they do not demand a stronger PD transistors for improving RSNM. Nevertheless, its complexity and need for one more port for the reading are the main disadvantages. Also, the RAT of the 8T cell is greater than the 6T if only one fin is used in the RD1 transistor.

In terms of leakage power, the 6T and the 8T SRAM cells show a linear dependency to the supply voltage and almost an exponential dependence on-chip temperature (10x variation in the range of 30-110°C)

Area of the 6T cell with 3 fins in the PD transistor is approximately 14% greater than the nominal 6T cell and the area of the 8T cell is approximately 35% greater.

When compared to the previous published work on the FinFETs in previous technologies, we conclude that moving to a smaller technology node (i.e. 10nm) while also reducing supply voltage according to the ITRS predictions, the RSNM is stable, leakage increases a little, and WLWM is significantly degraded due to the lower supply voltage. While RSNM results are positive (the read stability is guaranteed), a reduction in supply voltage may compromise write-ability. Plus, leakage currents, while small in absolute terms, they are exponentially dependable on temperature.

Is it a stockbroker? Is it a quantity surveyor? Is it a church warden? No! It's Bicycle Repair Man!

Monty Python, Bicycle Repair Man

5

Gain Cells

5.1 Introduction

The analysis from the Chapter 4 has shown that effects of the process and environmental variations are not entirely alleviated with the introduction of FinFETs, particularly in the small 10nm technology node. As it has already been shown, these issues reduce the cell stability (WLNM can be significantly reduced) and increase memory leakage (especially for higher temperatures).

Traditionally, most of the energy consumption of the semiconductor devices was spent on switching activity (e.g. dynamic energy) while leakage remained minuscule. Designers were focused on reducing dynamic energy consumption for decades. However, in sub 20nm devices leakage currents (and by that, static energy consumption) increases significantly. Process variations have even worsened the situation, and as a consequence of that imperfection, threshold voltage variability occurs. Accordingly, alternative cache architectures have to be investigated.

Multiple transistor DRAM cells are receiving significant attention lately for large embedded cache applications. These cells typically use the stored charge to control the transistor in the read-out path which is their major

difference when compared to the conventional destructive-read 1T1C memory cell. Thus, they have a non-destructive read [38]. The area and the leakage of these cells are reduced comparing to the classical 6T SRAMs while the read access time is not significantly degraded. Read access time can be increased by increasing the width of the read transistors because there aren't strict constraints that have to be satisfied as it is the case for the SRAM cells [28].

In this part of the thesis, we present the dynamic 3T cell built in the 10nm SOI FinFET technology. The cell is scaled to the minimal dimensions while keeping performance (i.e. speed). We analyse the cell access time, retention time and static power when exposed to the different sources of process variability (LER, FER, GER) and environmental variations (temperature).

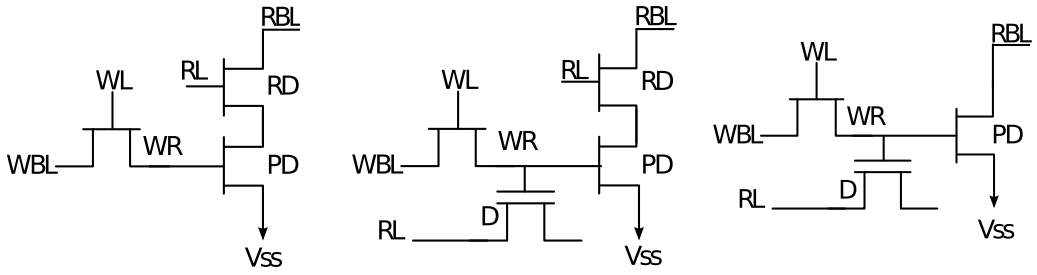
The main contribution of this work is that, as far as we know, this is the first that characterises the 3T cell for FinFET technology. We make a thorough analysis of the 3T cell and show that a read access time close to the 6T SRAM cell can be achieved. We further enhance the cell retention time and consider the effects they have on a large memory array implementation. Finally, we compare our work with similar solutions regarding DRAM cells and explain the advantages and disadvantages of our proposal.

This work was previously published in the "International Conference on Computer Design" (ICCD) in 2012 [72].

5.2 Related Work

Multiple transistor DRAM cells have received considerable attention lately as a replacement for the classical 6T SRAM cell. These cells use the capacitance of the device to store the memory state. The charge that flows through the read line is greater than the charge used for storing the value, so these cells are also called gain cells. A separate read path enables non-destructive reads and reduces the access time when compared to the traditional 1T1C cell. Also, when compared to the 6T SRAM, the different read line isolates the cell core from the read path, and there is no critical read noise margin to consider during the reading process. The issue is alleviated in the 8T SRAMs by having a separate read path but on the cost of the area and complicated

routing. Nevertheless, in any SRAM, device mismatch due to the process variations compromises the stability of the cell.



(a) 3T (b) 3T1D (c) 2T1D

Figure 5.1: DRAM Gain Cells

Significant work has been done lately regarding these cells. In [38, 36], Luk et. all proposed a novel 3T1D DRAM cell (Figure 5.1b), whose access time is comparable to the 6T SRAM. The gated diode acts as a storage device and an amplifier for the cell voltage. Before a read occurs, the RBL is pre-charged to V_{dd}. When a logic "1" (high voltage) is stored in the cell, the gate of the PD transistor is strongly biased during the reading process. High current flows through the PD transistor and the output capacitance of the RBL is forcefully discharged to the ground. On the other hand, when a low voltage is stored in the cell, the bias on the gate of the PD transistor is lower, the current through the PD transistor is decreased and the output capacitance is discharged slowly.

The same authors proposed in [37, 38] a 2T1D cell (Figure 5.1c) which uses the same idea of gated diode. The major difference is that no additional RD transistor is used. Instead, the source of the PD is connected to a positive supply, V_{ss}, which holds the PD transistor off when a logical "0" is stored in the cell even during the read process.

In [73], Bhoj et. al presented different architectures for 3T/2T1D DRAM cells for the 30nm Independent-Gate and Tri-Gate FinFET structures. According to their work, a 3T1D memory cell with Tri-Gate FinFET can be implemented with the speed close to the 6T SRAM and the leakage reduced more than 10 times but the area of that cell is almost 90% of the 6T SRAM and retention time is $4.2\mu s$. According to that work, the diode requires a big

gate to achieve good results (gate diode length is more than 20λ). Similar conclusions are drawn in [74] but for the 65nm bulk technology.

Apart from the speed, the central issue in these cells is the retention time (τ). Due to the leakage currents (mostly of the WR transistor) the cell loses its state after some time, and it has to be refreshed. In [75] authors show that 1% of processor performance is lost when the 3T1D with retention time of $2\mu s$ is used in the L1 cache.

Comparing to the cells that use the gated diode for storing the value, the 3T (Figure 5.1a) cell has two significant advantages:

- It does not need an additional diode. Thus, it saves area. Instead of that, it uses the gate capacitance of the PD transistor for storing the value. This implies that the cell area of the 3T cell is considerably reduced with respect of the 3T1D.

- Using the 3T cell in the memory design opens some possibilities to combat process variability at run-time. In the next section, we explain how the cell retention time can be increased. According to our study, applying these techniques to the gated diode cells is not effective since the noise margin (i.e. the difference between reading a logical "0" and a logical "1") is degraded as a consequence of the process variations.

5.3 Enhancements of the 3T DRAM cell

In this Chapter, we focus on the following four design goals: small area, high retention time, reduced leakage power and the speed comparable to the 6T SRAM. Accordingly, in this section we propose the techniques for improving cell performance.

Also, all these demands should be achieved when the effects of process and environmental variations are considered in the circuit analysis. In other words, all techniques that we propose do not compromise the memory yield that is achieved when the cache memory is implemented in the SRAM.

5.3.1 Retention Time Enhancement

Recent studies show that the operating temperature of the chips can be more than 90°C [76]. Knowing the fact that the transistor leakage is exponentially dependent on the chip temperature and that the retention time of these cells is directly dependent on the leakage of the WR transistor, it is of crucial importance to increase the retention time to the highest possible value.

Retention time of the cell can be increased in 2 ways:

- Applying a negative voltage on the WL pin when the cell is not in the writing process (i.e. idle or during a read). The retention time increases almost 10x per 0.1 voltage reduction. This method is limited by the following issues that arise: (i) the power consumption (static and dynamic) of the word line can increase slightly; (ii) stronger transistors for generating higher currents may be needed (thus, the increase the area of control/peripheral logic).

- Increasing the low-level voltage on the WBL. This reduces the drain-source bias of the WR transistor and in consequence, it reduces the leakage when a logical "1" is stored in the cell. However, this reduces the noise margin. The method is limited by the threshold voltage of the PD device (i.e. reading "0" state can be compromised).

According to our simulations, the low-level voltage of WL ($V(WL_{low})$) can be reduced around -0.2V. Applying the second technique on a 3T1D (e.g. increasing the low level up to 0.15V) cell is very hard since the noise margin between a logical "0" and a logical "1" can be significantly reduced. This is the consequence of two facts:

- Threshold voltage (V_{TH}) of devices that we use is between 0.15-0.2V.

- High variability of the I_{on} current. The difference between the minimal and the maximal value is more than 40%[6, 49].

On the contrary, the 3T cell does not have a diode and the gate of the PD transistor is not swung higher during the reading process. This does not

compromise the reading a "0" state when it is increased near the threshold value. Nevertheless, the lack of the gated diode increases the cell access time. The absence of the gate diode is compensated by making the PD transistor slightly stronger. The absence of the gated diode in the 3T cell makes the cell layout simpler and the overall cell size smaller. [1]

5.3.2 Periphery Circuit Re-Design

The negative side of the previously described techniques is that they demand additional power supplies. Besides the default V_{dd}, up to two additional voltage levels have to be provided: $V_{low}(WBL)$, $V_{low}(WL)$.

The design of the control logic that generates the negative $V_{low}(WL)$ when the cell is not accessed is the most difficult. Connecting a negative power supply instead of the GND level increases the leakage of the control circuit. On the other hand, reducing the $V_{high}(WL)$ voltage below V_{dd} increases the logic delay since the strength of the transistors is reduced. Our simulations have shown that the delay of the circuit is 15% larger when it is supplied with 0.7V and -0.2V comparing to the one that is supplied with the 0.8V. However, the writing process of the memory is usually not in the critical path. According to our simulations, a memory array of 128 rows, designed to function at 2.5GHz never fails in writing the cell when the WL is supplied with 0.7V and -0.2V. Also, apart from the voltage generators/regulators, the dynamic power of the control logic increases less than 1%, and the leakage stays the same. Additionally, the circuit delay can be reduced by increasing the strength (fin number) of the transistors at the cost of the area. But this has not been taken into account in this work as the cell already meets the target frequency.

While increasing the design complexity through the use of multiple volt-

[1] In [72] we suggested reducing the V_{ss} voltage for increasing the cell Read Access Time to make it equal to the 6T SRAM cell. However, in those simulations the channel length of the transistor was wrongly set which caused the greater RAT of the 3T cell. After correcting those errors in the simulation, it turns out that the RAT of the 3T cell is equal to the 6T SRAM when the parameters are according to the Table 5.1. With respect to this, applying a negative voltage to the V_{ss} is unnecessary, and it will not be considered in this thesis.

ages, the goal of this work is to show the potential of the 3T cell in the future technology scenarios. We believe that certain design complexity must be traded-off for better performance (delay, area, power and reliability).

5.4 Simulation Results

In this section, we present the simulation results of the 3T cell. We run the experiments for different operating conditions (i.e. different supply voltages and temperatures).

Table 5.1: Transistor dimensions of the 3T cell

	6T SRAM			3T DRAM		
	PU	PG	PD	WR	RD	PD
$L_g[nm]$	10	10	10	30	10	10
N_{fin}	1	1	2	1	1	3

The cell is sized for the minimal dimensions with one fin for the RD and the WR transistor and 3 fins for the PD transistor. The transistor dimensions are according to the Table 5.1. Length of the RD and the PD transistor equal to 10nm (minimal possible).

In order to increase the cell retention time, we implement the WR transistor with a higher channel length. Increasing the channel length of WR transistor will not degrade it's strength significantly (and by that, neither write speed) since carrier velocity of sub 20nm devices is limited by source-injection velocity, showing a weak dependence on the lateral field (i.e., V_{ds}/L). Also, V_{th} roll-off due to the influence of V_{ds} (Drain Induced Barrier Lowering) is much lower for FinFETs than in bulk devices. On the other hand, leakage will be significantly reduced (sub-threshold slope is strongly dependent on T_{FIN}/L ratio). Also, I_{off} has an exponential dependence on V_{th} (which depends on L). Altogether, increasing the channel length of the WR transistor will reduce leakage significantly while write speed will not be compromised. Also, in order to increase retention time further, we apply a negative voltage on the WL line when the cell is not accessed. In the rest of this work, we will consider a channel length of WL transistor of 30nm

When compared with the 6T FinFET SRAM cell that has 2 fins in the PD transistor (according to the [4] this is the optimal parameter for the Tri-Gate FinFET SRAM) the layout of the 3T cell is reduced approximately 40%.

The nominal supply voltage of the memory is $0.8V$. Process variability was simulated according to [49]. Monte Carlo simulations were performed when H_{fin} and T_{fin} parameters are considered Gaussian variables with the relative standard deviations of 12% and 16% respectively. Since there is still little information regarding the actual FinFET fabrication process, in our analysis we considered only intra-die variability. Also because LER, FER and GER are primary sources of the variability, we found it entirely independent from fin-fin for every FinFET [11]. For every particular case, 1000 instances were simulated to achieve high confidence.

5.4.1 Read Access Time

We simulated one memory column that consists of pre-charge logic, the 3T cell and the write logic. Besides the strength of the access transistors, the cell access time strongly depends on the equivalent bit line capacitance. This capacitance is mainly caused by the parasitic capacitance between the drain and the source of the access transistors, as well as, routing lines. Consequently, it is considered the same for the 3T and the 6T cells. The equivalent RBL capacitance is proportional to the number of rows of the memory array. In our analysis, we assumed $C_{RBL} = 5.12fF$, which we found to be equivalent to a memory array of 64 columns. Since there is still little information regarding the actual FinFET fabrication process, this value should be taken with little reserve. However, as the bit line capacitance is the same for the 3T, and the 6T arrays the relative results shown and comparisons made to the SRAM cells are not compromised.

We define the access time as the time between the moment when the RL reaches $V_{DD}/2$ until the RBL reaches $5V_{DD}/8$ (approximately $V_{DD} - V_{TH}$) when a logical "1" is previously written in the cell.

The following figures present the results for reading access time for different temperatures. Besides mean value, an error range is shown, also. The

error range is equal to $\pm 1.96\sigma$.

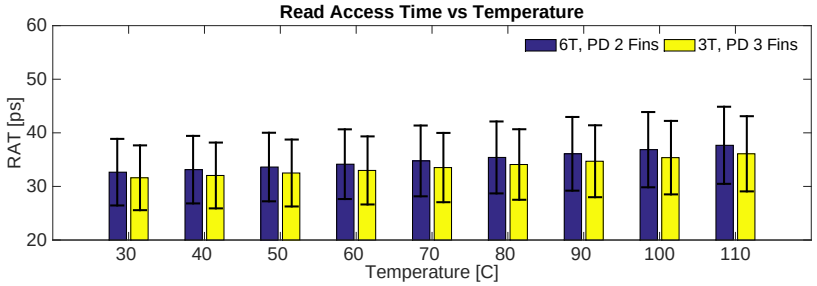


Figure 5.2: Read Access Time vs. Temperature

Figure 5.5 shows how the cell access time changes with the temperature. For comparison, the RAT of the 6T SRAM cell is presented in the figure, too. It can be observed that due to the stronger PD transistor, the 3T cell is even little faster than the 6T cell. Similar dependency on the temperature and the error range to the 6T cell is also evident from the figure.

5.4.2 Retention time

The stored charge decreases over time as a consequence of the leakage through the WR transistor. Because of the process variability some instances lose their state sooner than the others but refresh time must be defined according to the worst case. We define Retention Time as the moment when the "worst" cell read access time increases 5ps from its nominal value (that is measured from the first cycle after writing in the cell). Retention time can be defined also as the value when the potential on the gate of the PD transistor reaches certain value, but we have chosen this approach because we are interested in the cell which access time is approximately same as for the 6T SRAM.

Figures 5.3 and 5.4 show the Retention Time dependency on $V_{low}(WL)$ and $V_{low}(WBL)$. High variability can be noticed between the different values of $V_{low}(WL)$ (more than 3 orders of magnitude). Retention time increases more than 10x per 0.1V $V_{low}(WL)$ reduction. The Retention Time slope for different $V_{low}(WBL)$ is slightly later. In this work, we do not assume a $V_{low}(WL)$ higher than 0.1V because the higher voltage would significantly degrade the noise margin when a logical "0" or "1" is written.

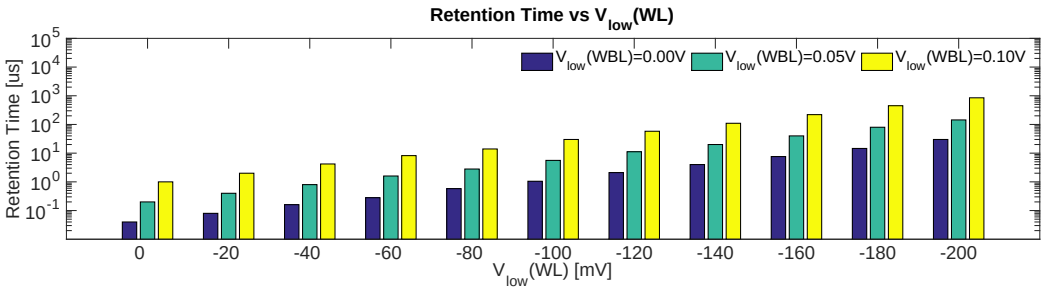


Figure 5.3: Retention Time vs. $V_{low}(WL)$

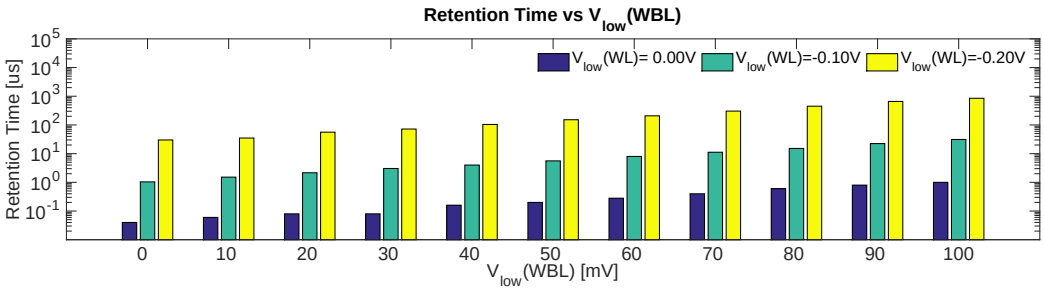


Figure 5.4: Retention Time vs. $V_{low}(WBL)$

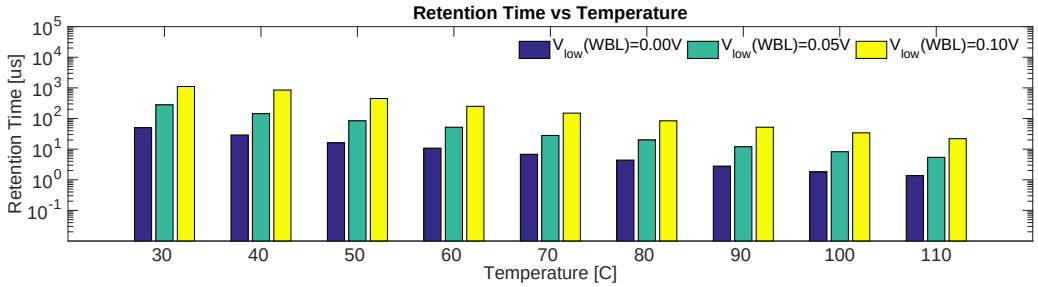


Figure 5.5: Retention Time vs. Temperature

According to Figure 5.5, the retention time shows very high temperature dependence. Due to the leakage of the WR transistor, the retention time varies more than 10^4x between $30 - 110°$C. However, even in the most extreme case of the temperature 110°C, retention time of $20\mu s$ is achieved for particular combinations of $V_{low}(WL)$ and $V_{low}(WBL)$.

Combining these two techniques should be considered when higher retention time is needed. The general rule is that a larger cache demands a higher retention time since the data life in these caches is longer. Accord-

ingly, we conclude that if the cell is implemented in L1 caches, good results can be achieved even with one additional voltage source (assuming a 1% performance loss for a retention time of $2\mu s$ [75]). However, if the cell is used in higher L2 and L3 implementation one more voltage source is needed to sustain the high retention time.[1]

5.4.3 Static Power Consumption

Leakage power is one of the most important parameters in the memory design. Figure 5.6 shows the leakage of a 4KB memory block when it is implemented using the 6T SRAM and the 3T DRAM. Same methodology for evaluation is used as in the Chapter 4. It can be seen that our proposal reduces leakage more than 7x.

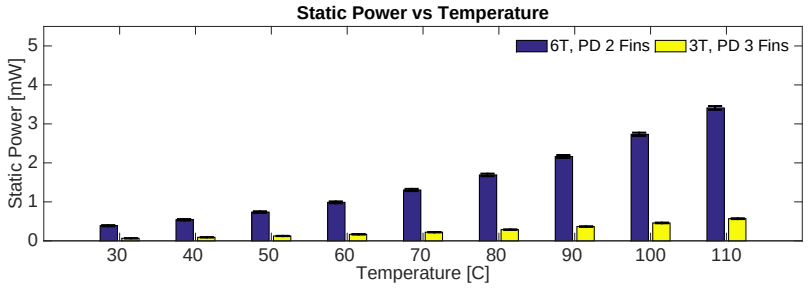


Figure 5.6: Static Power Consumption vs. Temperature

Here it should be noted that although the leakage is reduced more than 7x for a particular case, the memory dynamic power may increase because of the refresh policy (if any, [75]).

In [75], the authors present several refresh techniques for an L1 cache based on the 3T1D cell. According to their work, the 3T1D cell cache memory dynamic power can increase 1.3-2.25X comparing to the 6T cell memory depending on the refresh technique applied. Nevertheless, they also show that, when no refresh is used, the impact on the performance (IPC) is minimal. Because higher retention time can be achieved with the FinFET devices, higher refresh intervals and lower dynamic power overheads are expected.

[1] Alternative is to use weaker devices since these caches do not demand very low Read Access Time. However, in this work we used only predictive HP devices.

Additionally, the leakage power tends to dominate cache structures in the sub 22nm technologies and, thus, even if the dynamic power is increased, we can expect a reduction in the overall cache power.

5.4.4 Cell layout analysis

Following the same rules as in Chapter 4 we sketched the layout of the dynamic 3T cell. Figure 5.7 presents the layout of the 3T cell. When compared with the layouts of the SRAM cells presented in Chapter 4, significant area savings can be observed. The 3T cell is smaller approximately 32% than the nominal 6T SRAM cell, 40% smaller than the area of the 6T cell with 3 fins in PD transistor and almost 50% than the 8T cell.

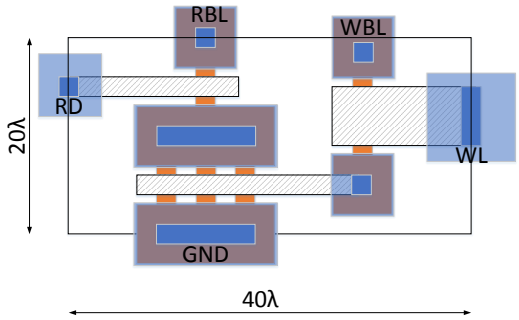


Figure 5.7: 3T cell Layout

5.5 Conclusion

In this Chapter, we presented the 3T memory cell with the speed close to the 6T SRAM for future 10nm FinFETs. We simulated the behaviour of the cell under the process and the environmental variations.The process variations are modelled at device level according to the ITRS prediction. We further propose to enhance the cell retention time by either (i) applying a negative voltage to the WR transistor when the cell is idle or (ii) by increasing "0" level voltage level on the WBL. The advantage, of applying these techniques, is that they can be implemented dynamically after chip fabrication. Thus,

they become an effective way to combat the process variability and to find optimal post-fabrication trade-off between the retention time, the noise margin and the write access time. On the other side, they require additional supply voltages that complicate the design.

High read access time can be achieved (close to the 6T SRAM) by combining these techniques. Retention time, although highly dependent on the chip temperature, can reach more than $20\mu s$ in the extreme conditions of 110°C. Also, the static power is reduced more than 7x and the area more than 32% when compared to the nominal 6T SRAM cell.

Well sir, I have a silly walk, and I'd like to obtain a Government grant to help me develop it.

Monty Python, Ministry of Silly Walks

6

DRAM Coherent Caches

6.1 Introduction

The dynamic gain cells that are introduced in Chapter 5 show high potential to replace the classical 6T SRAM cell that was the primary choice for high-speed cache implementation over the years. The main positive features that characterise these cells are an access time similar to the 6T SRAM cell, the reduction in static power consumption that can be more than 7x; and the cell area that is 40% smaller than the SRAM.

However, the major drawback of these cells, like any other dynamic cell, is that they demand refresh for keeping their value. Refreshing increases the dynamic energy and produces a performance loss since the memory access has to be blocked during that process. These effects are even exacerbated due to the minuscule retention time that these cells can have due to process and environmental variations.

On the other hand, it is well known that the life of the data in a cache memory is smaller as the capacity of the cache is smaller. This fact has been used as a motivation for the proposal of different optimisation techniques

that try to exploit this fact for reducing the refresh energy, by refreshing the most relevant cache parts [75, 77].

Currently, we can find a lot of different cache architectures for different multiprocessors in the market. For instance, a typical Intel CMP Ivy bridge architecture consists of 2x32KB of L1 cache per core, 256KB L2 cache per core and 3-12MB of shared L3 cache [78]. On the other hand some of the processors that implement ARM architecture have small private L1 caches 32-64KB and one shared L2 cache with up to 2MB [79].

In this chapter, we explore DRAM-based coherence caches. We perform our analysis when caches are designed for the predictive 10nm SOI FinFETs that we introduced in Chapter 3. We analyse the memory energy consumption and the overall system performance when the cache is implemented using the 3T gain cells. The ageing of this memory has been studied, too. We propose novel refresh techniques based on the coherence state of each cache line that reduce refresh energy and prolong cell life. In short, the main contributions of this work are:

- The proposal and the evaluation of DRAM-based L1 and L2 coherent cache hierarchy for two cache coherence protocols (MESI and MOESI) under the process and the environmental variations;
- Extensions of the MESI and MOESI coherence protocol to support DRAM refresh implementation;
- Performance figures based on the future SOI FinFET technology analysis;
- A dynamic mechanism to find the optimal refresh policy that requires the minimal refresh energy while keeping the system performance above some predefined level.
- The evaluation of the ageing of the 3T gain cell and the performance figures of how the coherency based refresh can be exploited for reducing memory ageing.

This work was published in the Design Automatisation Test Europe (DATE), in 2014 [80]

6.2 Related Work

6.2.1 Refresh Energy Reduction in Dynamic Memories

Many techniques, for reducing DRAM, refresh energy have been presented in the past [81, 75, 82]. Although the block refresh is the simplest solution for the implementation (refreshing the whole memory block after a predefined period), it doesn't leave space for any further optimisation.

On the other hand, techniques that assume line refresh are slightly complicated to implement but a significant amount of refresh energy can be saved. Accordingly, many proposals that can be found in the literature try to leverage this fact by finding the optimal method that will reduce refresh energy without hurting system performance very much. In the following text, we present some of the most significant works on this topic.

In [75, 82] authors report 2% of processor performance loss when a 3T1D cell with retention time (τ) of $0.8\mu s$ is used in the high-speed L1 cache when no refresh is applied. Although retention time of this memory is very low, this small performance loss is achieved due to the small memory size (baseline of the L1 cache size is 64KB).

Using a slightly different dynamic gain cell, the authors in [77, 83] analyse the performance and the power of the DRAM-based last-level shared L3 cache. They conclude that the refresh energy is the biggest contributor to the overall energy consumption. On the other hand, if no refresh is applied in the L3 cache, the performance is reduced almost by 40%. Additionally, they assume a cell retention time of $20\mu s$ which might be a challenge to achieve in sub 20nm technologies for the high-performance devices, especially at high temperatures. For instance, in our work in [72] and in Chapter 5, we show that in order to achieve a retention time of $20\mu s$ for the 3T cell for the 10nm FinFET for higher temperatures two additional voltage sources have to be used. This, of course, goes at the cost of the design complexity.

In this proposal, we consider the dynamic cache memories implemented in the 3T cells which we consider as a good compromise between the 1T1C and the 3T1D. Area of the 3T cell is smaller than the 3T1D, and read access time is not significantly lower [72].

Many papers explore refresh techniques for the different memory levels. Refreshing the memory rows in different time periods based on the cell retention time is presented in [84]. Lines are refreshed based on their retention time estimated in an initial profiling, and a 72% of the refresh energy reduction is claimed. Refreshing a memory line based on the data importance that is defined by the programmer is presented in [85]. Refreshing the line based on the last access [81] and the delay refresh period by implementing the line counters (52% savings for refresh energy in 2GB DRAM).

Some, on the other hand, focus on the improving DRAM performance by dynamically finding the optimal refresh intervals in order to prevent collisions with possible memory reads and writes [86, 87, 88].

In [89], authors present intelligent refresh techniques for the L2 and L3 caches in order to reduce the refresh energy. The decision to refresh is based on the time elapsed since the last access, and its state (clean, dirty, idle-dirty for lines not being accessed for a long time). They claim 56% energy reduction comparing to the classical SRAM implementation with 6% increase in execution time. However, this work is based on low power devices and a nominal system frequency of 1GHz in order to sustain retention times up to $100\mu s$.

In [90], the authors show how the spatial locality of the process variability can be exploited to reduce the refreshing energy of large eDRAM caches. They claim that if the spatial locality of process variation exists, the majority of cache lines can be refreshed with significantly higher period. That reduces the refresh energy by 43%.

The above papers use line-based refresh counters. These counters are set at different pre-defined intervals. None of the papers above uses the coherence state as the criteria to refresh. None of these works run simulations for the future 10nm FinFET technology. Plus, most of these works do not consider the implementation of coherent caches in DRAM technology (and

the consequent changes to the coherence protocol).

In this Chapter, first we propose how the cache coherency can be exploited as a proxy to determine if a line in the cache has to be refreshed or not for reducing the refresh energy while keeping system performance. We define how the cache coherency protocols (MESI and MOESI) have to be modified in order to support this implementation, and later we present the dynamic approach to determine the optimal refresh policy (based on the coherence state) that reduces the refresh energy while keeping execution time below the predefined threshold. As far as we know, this is the first time that cache coherency is used for reducing dynamic memory refresh energy.

6.2.2 BTI Aware Design

As previously said in Chapter 1, the variation in the device parameters due to the device ageing is a significant problem that should be considered very carefully during the design process. Addressing the issue of device ageing (BTI) on the micro-architecture level has been already investigated in the past. In this section, some proposals are presented. Although some of the papers are orthogonal to our approach, we report them here so that the reader can get some insight into the numbers how much life of the chip can be extended when the BTI aware design is applied.

In [91], authors present the design of an NBTI-aware processor. They propose strategies to mitigate the degradation in the combinational and storage blocks. They propose to turn off idle parts, reduce the voltage level, balance data probability in memories, etc. They reported up to 10X reduction of the V_{TH} ageing (reduction to just 2% comparing to nominal 20%). Similarly, particular circuits for automatic flipping the value of a cell are presented and analysed in [92].

In [93], the authors proposed an NBTI aware design framework for multi-core systems. They developed an algorithm to balance workloads among different cores. Their results for the 64 cores systems show a reduction of the core failure rate by 20% and the MTTF by 30% with a small degradation in the system performance of 6%. Similar work is presented in [94].

The ageing of a multicore system is also investigated in [95] where voltage scaling is used to relax the most stressed cores. The technique tries to distribute homogeneously the ageing among the cores. They report the extension of the chip life by 2 years.

Finally, in [96], authors show that the life of the system can be extended by gradually increasing the voltage over years. They report 46% increase in the lifetime.

6.2.2.1 Gain Cell Ageing

Compared to the other papers, this work is focused on the evaluation and the reduction of the gain cell degradation over time. We concentrate on the cell degradation mitigation by using the cache coherency state of each line, and we just refresh lines in a a subset of the states. Our analysis is based on the 3T cell, but the same principle can be applied to any other gain cell. Also, since this cell is designed in NMOS transistors, the only visible BTI effect is PBTI.

In the literature, one can find proposals of gain cells designed with PMOS transistors. For instance, in [77], authors present a gain cell designed with the PMOS transistors. The biggest advantage they have is the reduction in leakage energy and, as a consequence, a higher retention time. On the other side, the read access time is greater.

The methodology to estimate the cell ageing that we present in this work can be applied as is for the PMOS ageing, too. Also, according to [13] device ageing is approximately the same due NBTI and PBTI for the 10nm FinFET devices. Consequently, we expect that the degradation results we present here for the NMOS gain cell should be very similar to the PMOS gain cell.

6.3 Cache Coherency Protocols

6.3.1 MESI protocol

The MESI protocol [97, 98, 99, 100, 101] is a widely-used cache coherency and memory coherency protocol. Every cache line is marked as one of the following states:

- **Modified (M)**: Cache line exists only in that cache, and it is dirty (it is not consistent with the lower level memory value). It should be written to lower memory level before it is invalidated or replaced.
- **Exclusive (E)**: Cache line exists only in that cache, and it is clean (it is consistent with the value in the lower level memory).
- **Shared (S)**: Cache line may exist in some other cache, and it is clean.
- **Invalid (I)**: Cache line doesn't hold valid data.

A cache read can be serviced from any cache state except Invalid. On the other hand, cache writes can take place only if the line is in Modified or Exclusive state. If the cache line is in Shared state, first we have to trigger a request for ownership, and, consequently all shared lines will be invalidated first.

A cache line that is in Modified or Exclusive state has to "snoop" all the other caches accesses to intercept any request to the same address in the lower memory level. If this is the case, the line changes to the Shared state, and if the line was previously in the Modified state, it is written back to the lower cache before moving into the Shared state and sent to the requester.

6.3.2 MESI extensions for DRAM support

In this section, we show how the cache coherence state can be exploited to find an energy-delay optimal refresh line policy. Since different programs access cache in a different way, we propose coherency-aware refresh. In other words, we consider refreshing a cache line depending on its coherence state.

In order to make this possible, we extend the MESI protocol the following way:

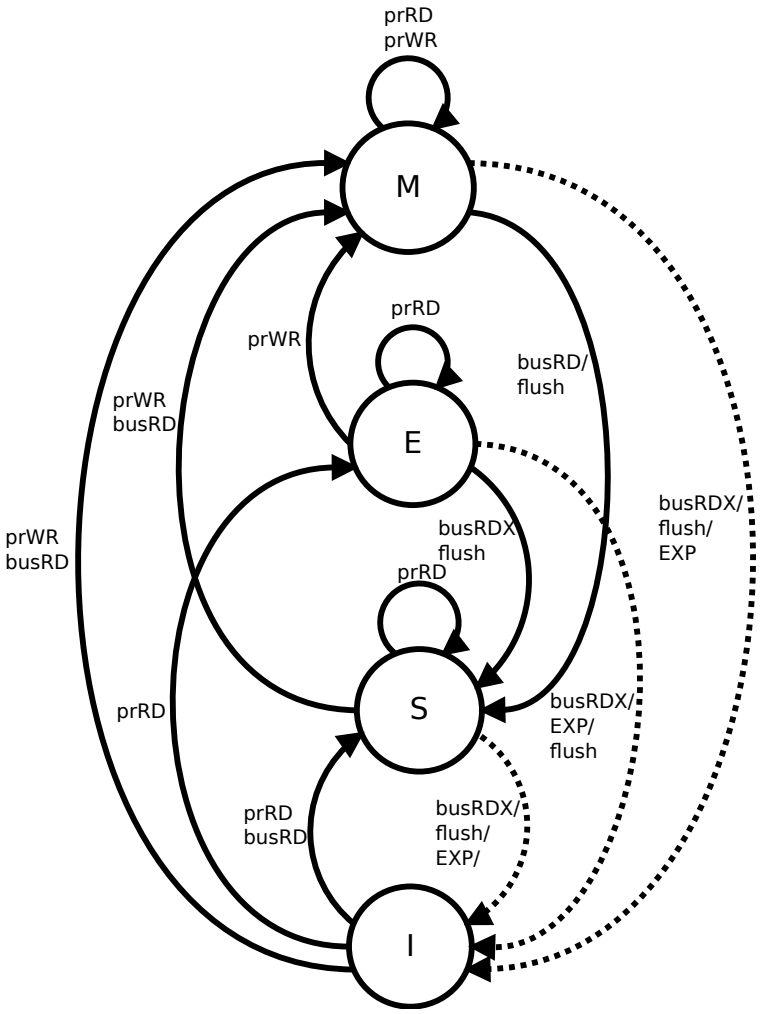


Figure 6.1: MESI protocol with the extensions (dashed lines) to handle expired lines

- **Modified (M)**: In a case when a Modified line expires, and depending on the explicit refresh scheme, it should be either refreshed which keeps the line in Modified state or it must be evicted to a lower memory level before it is tagged as Invalid.
- **Exclusive (E)**: If the line is refreshed, it will stay in Exclusive. Otherwise, it must be invalidated (no eviction is needed since the correct

data exist in lower cache).

- **Shared (S)**: If a cache line in this state is refreshed it stays in Shared. In the case of no refresh, the line must be invalidated (same as Exclusive).

- **Invalid (I)**: Since this line doesn't hold valid data it should never be refreshed since that would be unnecessarily energy consumption.

Table 6.1: Refresh policies defined for MESI protocol

Policy	Modified	Exclusive	Shared	Invalid
BLOCK	x	x	x	x
MES	x	x	x	
ME	x	x		
MS	x		x	
ES		x	x	
M	x			
E		x		
S			x	
NONE				

According to the combination of the line states that can be refreshed or not, 8 refresh policies can be defined: NONE, M, E, S, ME, ES, SM, MES. Letter defines that line in a particular state is refreshed, NONE means that no line state is being refreshed (Table 6.1).

Figure 6.1 presents the state diagram of modified MESI protocol. Dashed lines show modified transitions. Standard line definitions are used to describe state transitions ("PrRd" - Processor Read (Read request from processor), "PrWr" - Processor Write (Write request from processor), "BusRd" - Bus Read (Read request from the bus without intent to modify), "BusRdX" - Bus Read Exclusive (Read request from the bus with intent to modify)). Label "Exp" is used to define expired line transition.

6.3.3 MOESI protocol

MOESI is a directory based cache coherency protocol. According to [102] each cache line can be found in one of five states:

- **Modified (M)**: Cache line exists only in that cache, and it is dirty (it is not consistent with the lower level memory value). It should be written to lower memory level before it is invalidated or replaced.
- **Owned (O)**: Cache line may exist in some other caches, but it is dirty. A write to this cache line makes the Owner send a message to invalidate that data in any other cache. If some other core sends a request for the line, "dirty" sharing can be done (line is forwarded to that core without updating the lower cache level);
- **Exclusive (E)**: Cache line exists only in that cache, and it is clean (it is consistent with the value in the lower level memory). In the case of a write action from the same core, the line switches its state to Modified.
- **Shared (S)**: Cache line exists in other caches, and it can be clean or dirty. Some other cache is the owner. If the core wants to write this line, it has to request its ownership at first, and the former owner must set his copy to Invalid.
- **Invalid (I)**: Cache line does not hold valid data.

Comparing to MESI that is a snoop based protocol, MOESI is a directory based. The most simple implementation of the directory can be a simple bit vector for every line in the L3 cache (assuming that L3 is inclusive). One bit is reserved per each core. When the line is written in the private cache of the core, this bit should be set to '1'; otherwise it holds a '0'.

6.3.4 MOESI extensions for DRAM support

Due to the possible non-permanent nature of DRAM data when considering different refresh policies (i.e. other than "always refreshing"), the coherency

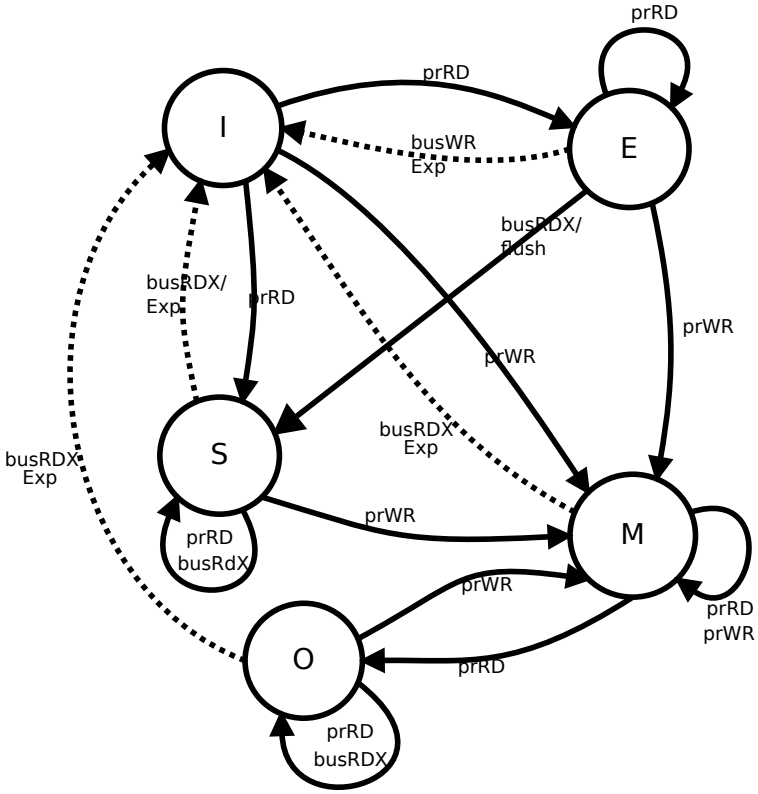


Figure 6.2: MOESI protocol with the extensions (dashed lines) to handle expired lines

protocol has to be extended to consider the cases when a line stops being refreshed, and data may be lost. Figure 6.2 shows an extended state diagram for the MOESI protocol. Dashed lines present the modified transitions that occur if the line expires and its state is not marked as refreshable (i.e. transitions to avoid any data loss).

- **Modified (M)**: In case that a Modified line expires, and depending of on the explicit refresh scheme, it should be either refreshed which keeps the line in Modified state or it must be evicted to a lower memory level before it is tagged as Invalid. The eviction to the directory has to be sent, too.

- **Owned (O)**: In case that an Owned line expires, and depending of on the explicit refresh scheme, it should be either refreshed which keeps

the line in an Owned state or it must be evicted to a lower memory level before it is tagged as Invalid. Directory has to be updated, too. If any of the shares wants to perform a write, first it has to request the ownership and, consequently, invalidate the line in all other caches. After this, the write can be completed.

- **Exclusive (E)**: If the line is refreshed, it will stay in Exclusive. Otherwise, it must be invalidated (no eviction is needed since the correct data exist in lower cache). Directory has to be updated, too.
- **Shared (S)**: If a cache line in this state is refreshed it stays in Shared. In the case of no refresh, the line must be invalidated (same as Exclusive). Although this line can be dirty, its value exists somewhere in the system (some other cache has this value in Owned state) so eviction is not necessary before line invalidation. Directory has to be updated.
- **Invalid (I)**: Since this line doesn't hold valid data it should never be refreshed since that would be unnecessarily energy consumption.

Because the MOESI has one more state than MESI, the number of refresh policies that can be defined is doubled: NONE, M, O, E, S, ME, MO, MS, OE, OS, ES, MES, MOS, MOE, EOS, MOES (Table 6.2). Initially, we may think that having more refreshing alternatives will increase the probability of finding an optimal refresh scheme that minimises the refreshing energy. However, as we will see when we discuss our dynamic algorithm, this has its negative sides too.

6.4 Dynamic Refresh Policy Determination

In this section, we describe the algorithm to determine the best refreshing policy dynamically. We present a detailed explanation of the MESI protocol, and later we indicate how it is extended to support MOESI cache coherence.

In the case of the MESI protocol, as in [80], we define 8 refresh policies according to the states where refresh is performed: NONE, M, E, S, ME, ES, SM, MES.

Table 6.2: Refresh policies defined for MOESI protocol

Policy	Modified	Owned	Exclusive	Shared	Invalid
BLOCK	x	x	x	x	x
MOES	x	x	x	x	
MOE	x	x	x		
MOS	x	x		x	
MES	x		x	x	
OES		x	x	x	
MO	x	x			
ME	x		x		
MS	x			x	
OE		x	x		
OS		x		x	
ES			x	x	
M	x				
O		x			
E			x		
S				x	
NONE					

The refresh policy can be changed dynamically in order to find the alternative that minimises refresh energy while keeping performance above some predefined threshold. A block diagram of the refresh policy determination system is presented in Figure 6.3, and Algorithm 1 presents its pseudocode.

To select the available refresh policies, we use two sets of counters. "IPC_{count} counters" that count the total number of committed instructions and "Refresh counters" that count the total number of refreshed lines during one characterisation period.

The selection of the refresh policy starts when the "StartCalibration" is triggered. In this proposal, this will depend on the moment defined for sampling but it could be triggered by any other criteria the designer may think about. Initially, the "Control Logic" selects the MES refresh policy. "IPC Counter" and "Refresh Counter" count the total number of the executed instructions (for every core) and the total number of the refreshed lines (for every cache memory), respectively for a fixed time period. "Con-

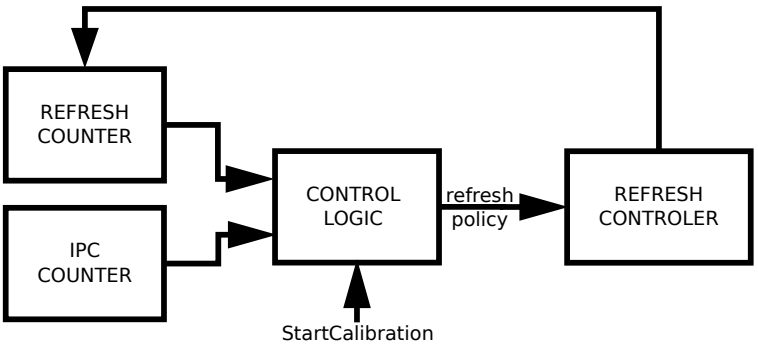


Figure 6.3: Dynamic Determination of Refresh Policy

if *StartCalibration* **then**
  $RefreshPolicy_{optimal} = MES$;
  $IPC_{optimal} = IPC_{count}$;
  $IPC_{threshold} = IPC_{count}x15/16$;
  $Refresh_{optimal} = Refresh_{count}$;
  for $RefreshPolicy$ **in** $RefreshPolicy_{set}$ **do**
    read IPC_{count}, $Refresh_{count}$;
    if $IPC_{count} > IPC_{threshold}$ **and** $Refresh_{count} < Refresh_{optimal}$ **then**
      $IPC_{optimal} = IPC_{count}$; $Refresh_{optimal} = Refresh_{count}$;
      $RefreshPolicy_{optimal} = RefreshPolicy$;
    else
      continue
    end
  end
end

Algorithm 1: Refresh Policy Determination Algorithm

trol Logic“ keeps track of the $IPC_{threshold}$ (e.g. 93.75% of IPC_{count}), the $Refresh_{count}$ and the IPC_{count} count as well as the optimal (i.e. best scheme so far) ($IPC_{optimal}$, $Refresh_{count}$), and switches to the next refresh policy from $RefreshPolicy_{set}$.

The control logic evaluates the next refresh policy, and we get new values of IPC_{count} and $Refresh_{count}$ for the same period. If the new IPC_{count} is below the calculated threshold value, the ”Control Logic“ automatically discards this policy and evaluates the next refresh policy. If the IPC_{count} is above the estimated threshold and the current $Refresh_{count}$ is smaller than

the best current refresh value ($Refresh_{optimal}$), then this policy is selected as the best. This is an iterative process. This process is repeated for all 8 refresh policies. Following this procedure, we will be able to select the policy with the minimum amount of refresh energy that has the IPC_{count} below some predefined threshold.

This is a very simple procedure that can be implemented either in the hardware or the firmware. If this function is implemented in the silicon the total logic needed for this implementation includes: 1 counter per core for IPC determination, one counter per cache memory for refresh number determination, and one adder for summing these values. Also, for control logic implementation 3 registers and 1 comparators are needed as well as simple state machine to control whole process. We assumed 32-bit logic (counters, registers, comparator and adder) as they provide enough resolution. When compared to the total cache memory area and power, this circuit provides a negligible increment.

The only difference in the implementation of the MOESI protocol is the larger number of the refresh policies defined (NONE, M, O, E, S, ME, MO, MS, OE, OS, ES, MES, MOS, MOE, EOS, MOES).

When MESI and MOESI protocols are compared, it should be noted that the execution time of this iterative sampling in the MOESI protocol is 2x higher since the number of alternative policies is doubled. However, in theory, the MOESI implementation can give slightly better results because the larger number of policies gives more possibilities to choose the appropriate one to minimise refresh energy.

The process could be even improved in some cases. For instance if the ME refresh policy does not pass the test (IPC is below certain threshold) then M, E, NONE polices should not been tested. This could be beneficial for the protocols with greater number of states (e.g. MOESI).

6.5 Simulation Results

6.5.1 Methodology

In this section, we present the simulation results. The system configuration parameters can be found in Table 6.3. For the simulation, we used marssx86, a full-system simulator for x86-64 CPUs [103, 104, 99].

We only simulated system with 2 cores with private L1 caches (Data and Instruction) and private L2 cache. L3 cache is shared. We assumed only 2 cores system because the workloads that we used in this work have very small footprint. It would be very hard to simulate system with higher number of cores (and accordingly greater cache size) since the most of the cache would be empty and the effects of our proposals cannot be seen.

In order to simulate the dynamic memories with marssx86, we enhanced it with the additional refresh controller logic. We implemented the line (or block) counters to measure the lifetime of each line in the cache (i.e. time since last write or refresh). The counters are 5 bits wide. When the counter reaches a certain limit (minimum retention time), the refresh controller is triggered, and it reacts according to the configuration (refresh policy). If the line is meant to be refreshed, it blocks a cache access port to complete the refresh (2 cycles in this work). In case that the line is not refreshed, the controller acts accordingly (as explained in the sections 6.3.2 and 6.3.4) in order to preserve data consistency. Also in our configuration, the tag and data arrays are both implemented in DRAMs, and both are accessed in parallel.

We consider the L1 and L2 levels dynamic while the L3 (that is shared in this configuration) we consider static because in this work we are focused on minimising energy of the coherent caches. By doing this, we prevented a possible influence of the L3 refresh on the coherent caches behaviour and the overall system performance. This configuration might seem against current memory trends and motivation because the lower memories are slower, greater and by that implemented in DRAM technology and the higher are smaller, faster and implemented in SRAM. However, we just want to prove

Table 6.3: Base System Architecture

Processor	2 core Out of Order, 2-wide issue width
L1 Data Cache	2x32KB 64 B line size 8 Ways 1 bank per cache Delay 2 cycles MESI/MOESI
L1 Instruction Cache	2x32KB 64 B line size 8 Ways 1 bank per cache Delay 2 cycles MESI/MOESI
L2 Cache	2x256KB 64 B line size 8 Ways 1 bank per cache Delay 5 cycles MESI/MOESI
L3 Cache	3MB 64 B line size 12 banks 16 Ways Delay 8 cycles Shared (write-back)
Main Memory	2GB (50ns delay) 1 channel
Technology	10nm FinFETs
Retention Time	3us
System Frequency	2.5GHz

that the idea of cache coherency based refresh stands, and to give some general baselines how this idea applies to any other memory configuration and size.

Although in [72] we report retention times above $20\mu s$ for the 3T cell, two additional voltage sources are needed for accomplishing that. This is, of course, a great downside for this proposal. On the other hand, if we look

at the impact of the retention time on performance, in [75] authors claim that high retention time for the L1 caches is not necessary (2% performance loss for $0.8\mu s$). Consequently in this work, we assume a baseline retention time of $3\mu s$, and we perform a sensitivity study later in this section. For the 10nm FinFET technology, this value can be achieved just with one additional voltage source. A small negative voltage in the WL has to be applied, when the cell is in hold mode, to reach this value at higher temperatures [72].

For our analysis, we used two parallel benchmark suites - PARSEC and SPLASH-2 [105, 106, 107]. All the workloads we configured as two threads. We used simmedium configuration for PARSEC. SPLASH benchmarks are set up as in [108]. The initialization phase is skipped, and only the application region of interest is simulated. All workloads run on top of the Ubuntu 9.04 (Linux 2.6.31). The full list of workloads that marssx86 supports can be found in [103]. All of them are simulated. We report 6 individual benchmarks plus the average (of all benchmarks in both suites). This sample was chosen to present the extreme cases, as well as the diversity in the behaviour for the different refresh policies. In other words, we tried to avoid plotting the benchmarks that show similar behaviour. Still, the mean column is the average of all the benchmarks.

6.5.2 Dynamic Algorithm Sampling Time Sensitivity

In this section, we present the procedure to determine of the minimal sampling time used in the algorithm. The key point of this discussion is to prove that the algorithm delivers the same final decision (i.e. refresh policy) when executing the whole application and when executing just for the sample period defined.

In order to determine the minimal number of instructions needed to deliver a correct result, we did the following experiment. We first run each application for a total of 1 billion of instructions (0.5 billion per core) for every refresh policy and the total execution time and the energy consumption were calculated for each refresh policy (results are normalised according to the system that has SRAM coherent caches). Those numbers were marked

as the reference.

Then, we iteratively run the whole benchmark suite for a smaller number of instructions (the step size is 50M instructions) until the final refresh policy choice was different than the reference one. We found this limit to be 150 million instructions (75M per core).

While this sets the minimum value used, we also evaluated a more conservative approach. This conservative approach sets a limit of the error of the performance and energy values measured to 5% or below. Using this method, the minimum number of instructions to be sampled is 300 million (150M per core).

Since we simulate only the representative part of each benchmark, and they are very regular, we just sample once at the beginning of the run. Testing several sampling intervals is left for future work.

6.5.3 System performance

Figure 6.4 shows the normalised execution time for the MESI-based caches. Results are normalised to the baseline multiprocessor that has SRAM coherent cache memories. Figures 6.4 also include a bar for the simple block refresh technique. This technique is the easiest for DRAMs: when the counter reaches the limit it refreshes the line regardless of its coherent state. Also, it assumes whole memory refresh (even the invalid lines) that would needlessly block the cache access and consume refresh energy.

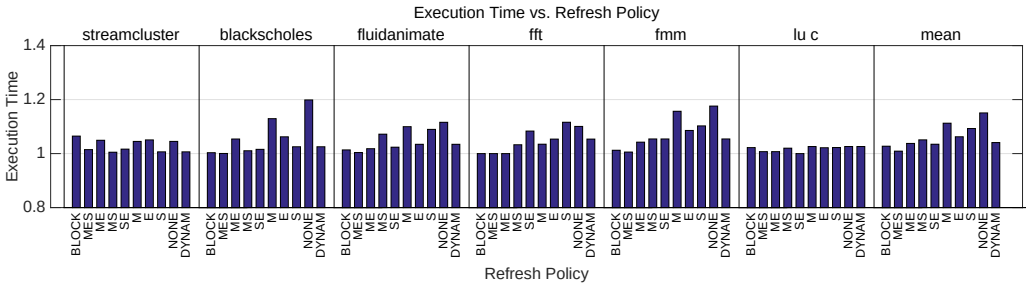


Figure 6.4: System execution time for different refresh policies normalised to the SRAM coherent cache for the MESI protocol

For the simulated system configuration and the available set of bench-

marks, results for MESI and MOESI protocol are very similar. The reason for this is that for these small caches, minuscule number of lines goes in the Owned state. So defining additional refresh policies for MOESI state didn't make much difference. In other words, if the Owned and Shared lines are summed in one for refresh policies definition, the results for MESI and MOESI are very similar. According to our simulation, those differences are less than 2% that is less than the error of the simulator that is stated in the [103]. Because of that, in order to avoid overcrowding of graphs that show almost same numbers, in the evaluation we present results just for the MESI protocol.

In general, the MES technique delivers the best performance. Performance is at the same level as the SRAM baseline. Different performance losses can be observed for different workloads for the various refresh policies. For instance "blackscholes" achieves smallest performance loss when the "Shared" state is refreshed, "fluidanimate" for the "Exclusive" state, while performance loss for "streamcluster" is very small for all refresh policies.

A fixed static scheme can yield significant performance loss, in contrast, the dynamic system can keep the performance loss below 4% for MESI. This is a tiny loss comparing to the overall energy benefits that we will present in short.

It is important to notice that the dynamic policy is based on the performance as well as on the refresh energy consumption. In terms of performance, the dynamic policy does not necessarily provide the optimal solution. This is because the algorithm only checks if the performance loss for the refresh policy is below a certain threshold. In other words, if the performance loss for two or more refresh policies is below the predefined threshold, the algorithm will choose the one with the smallest number of refreshes.

6.5.4 Energy Consumption

For energy estimation, we used simulation results that are previously obtained for the 6T SRAM and the 3T DRAM (Chapters 4, 5). We were also lead by the ITRS prediction[1] in order to estimate Dynamic Power. This

report gives prediction how the Dynamic power per cell is going to scale over the next 10 years. We assumed that the dynamic power per cell is the same for the 3T and the 6T cells. This is very realistic to expect considering the same technology node and the same access time as previously simulated. Similar assumptions were made in [77].

After the power estimation at the circuit level, a full system simulation was done with the MARSSx86 simulator. We used the total number of cache accesses (number of total read hits, read misses, write hits, write misses) and cache "snooping" communication in the coherence protocols in order to estimate the dynamic energy. We also extracted the total number of refresh lines in order to evaluate the refresh energy and the total number of cycles in order to evaluate leakage. We also included the energy consumption of the additional logic that includes: line counters and policy decision logic. Combining the access/cycle counts and the energy per access/cycle we obtained the final energy numbers.

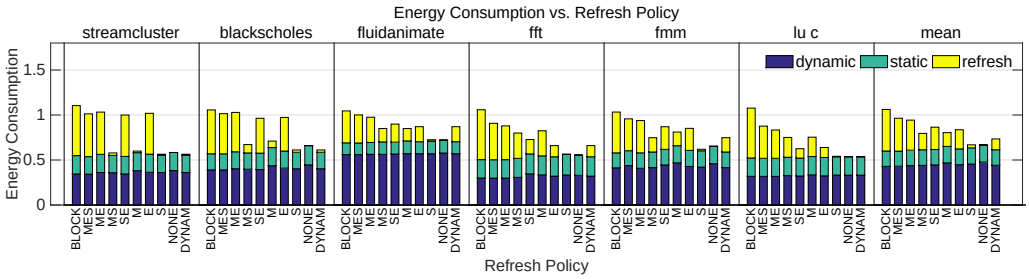


Figure 6.5: Energy consumption of the coherent caches (L1+L2) for different refresh policies normalised to the SRAM coherent cache for the MESI protocol

Figure 6.5 shows the total energy of the coherent caches (L1 and L2). The consumption of the line counters that are implemented to count the line retention time for dynamic caches is also included. The data are normalised according to the SRAM baseline. These figures show that the block refresh technique achieves the same energy consumption as the baseline. When our refresh techniques and the dynamic algorithm are applied, we produce significant refresh energy benefits. On average, the savings are 37% for the MESI protocol. According to the ITRS [1], the overall power consumption in

on-chip memory will reach the 50-60% of the total system power in a couple of years. Having this in mind and assuming the minimal performance loss showed, DRAM-based coherent caches show themselves as a good candidate to save energy in future designs.

It should be noted that an application of some refresh policies increases the leakage energy. This happens because the execution time extends for certain workloads for some refresh policies. However, the dynamic scheme ensures that this does not occur ever.

Also, the total system energy can increase due to the increase in the system execution time. However, when dynamic approach is applied, that guarantees execution time below some threshold (in our case mean value of system performance loss is 4% for all workloads) this enlargement is expected to be minimal.

6.5.5 Chip Communication

Given that the DRAM-based caches may induce a higher number of accesses to the lower levels of the memory hierarchy, Figure 6.6 shows the number of accesses for all caches (L1+L2+L3). It can be observed that the miss rate can increase for some benchmarks for individual refresh policy. However, when the dynamic approach is applied this rise can be neglected. This is reasonable to expect because the dynamic energy of the caches does not increase much as it has been shown in the previous section.

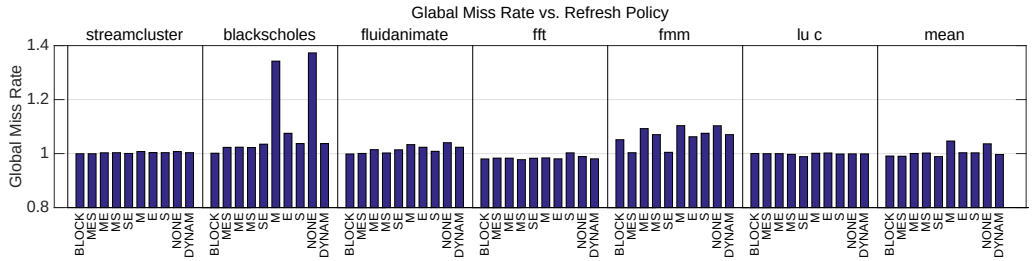


Figure 6.6: Miss rate for different refresh policies normalised to the SRAM coherent cache for the MESI protocol

We also observed the total number of accesses for every cache. We noted

that the total number of L1 hits dominate this number (in the baseline cache configuration) contributing with more than 99% of total L1, L2 and L3 cache accesses.

6.5.6 Retention Time Variation

As stated in the Chapter 5 the process and the environmental variations affect the 3T cells mostly in terms of the retention time [72]. The variation in the retention time exists due to the variation of the leakage drive current of the WR transistor when the cell is not accessed. This variation is caused by the process and temperature variations. In this section, we present how the variation in retention time affects the system performance and the overall energy consumption.

We wanted to illustrate the change of the system execution time and energy consumption for the different values of the retention time. Also, in order to avoid overcrowding of the figures, we show results only for BLOCK, MES, MS and NONE refresh policies and for the DYNAM approach. Very similar behaviour has been observed for other refresh policies.

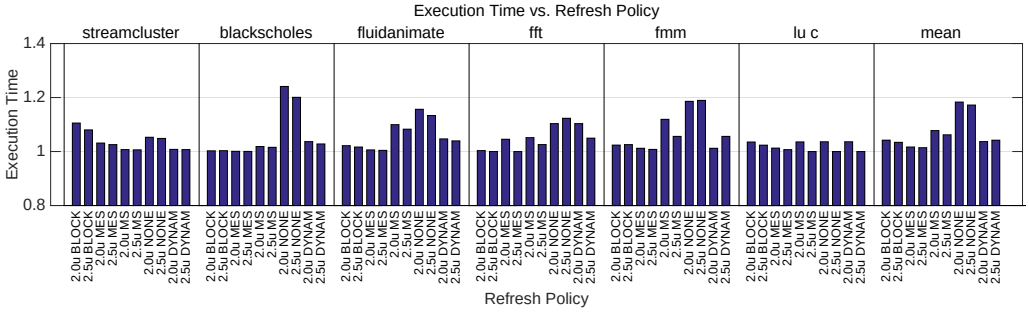


Figure 6.7: The system execution time for different refresh policies and retention times normalised to the SRAM coherent cache for the MESI protocol

We analysed the performance of the system when the retention time of the memory falls to 2.5us and 2us. Figure 6.7 presents the execution time results for the MESI coherence protocol. The results are normalised to the SRAM memory (i.e. refresh is not needed). Also, the energy consumption is presented in Figure 6.8. Some of the proposed refresh configurations can

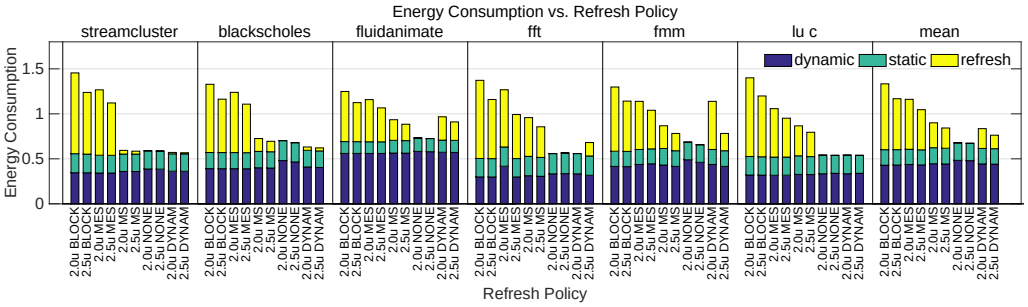


Figure 6.8: Energy consumption of the coherent caches (L1+L2) for different refresh policies and retention times normalised to the SRAM coherent cache for the MESI protocol

achieve similar results in the performance, and the refresh energy savings are higher (when compared with the BLOCK refresh) than for baseline retention time. Workloads ”streamcluster“ and ”blackscholes“ show good energy savings under the MS policy with significant energy reduction of up to 44% with minimal impact on performance even for the memory with 2us retention time. Other benchmarks such as ”fluidanimate“ show energy savings up to 18% and 24% respectively but the performance loss is around 7% (comparing to the SRAM baseline). Similar energy savings are achieved for ”fft” for the same policy with slightly smaller performance loss. On average, the MS refresh policy can save 25% of refresh energy with a performance loss of 9% (comparing to the SRAM baseline). On one end of the spectrum, energy savings can get up to 47% when none line is refreshed(i.e. NONE policy). Nevertheless, that configuration takes a 25% hit on performance. At the other end of the spectrum, even always refreshing (BLOCK) takes a performance hit, and it shoots up the energy consumption. When the dynamic approach is applied to the system, the total energy savings of the coherent caches are even higher comparing to the nominal BLOCK refresh (56%). For this simulation, the performance threshold was kept on the same level of 5%.

6.5.7 Temperature Dependency

Temperature affects the leakage dramatically. Plus, in the DRAMs, temperature has an extraordinary impact on the retention time. This impact on

the retention time may mean an increase in the refresh activity and consequently the overall energy and execution time. According to our HSPICE simulation of the 3T cell, there is a big reduction in cell retention time across the temperature range between $25°C$ and $80°C$. Approximately, we observed a decrease of 50% of retention time per $10°C$ for these devices. In order to present the best temperature effects for the different refresh policies, we show the results for two temperatures $60°C$ and $70°C$.

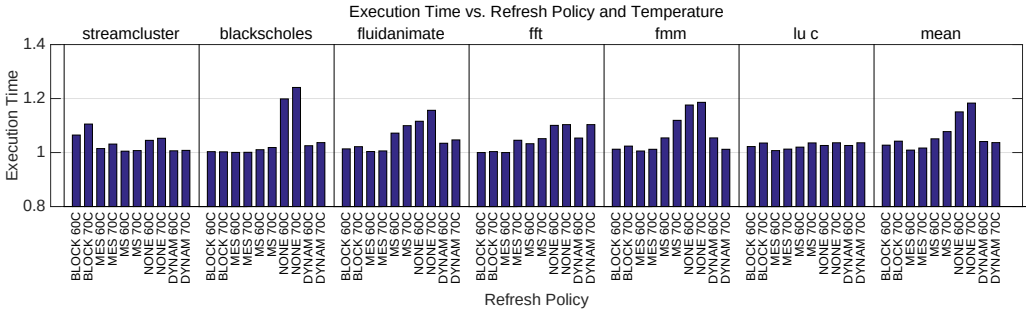


Figure 6.9: The system execution time for different refresh policies and temperatures normalised to the SRAM coherent cache for the MESI protocol

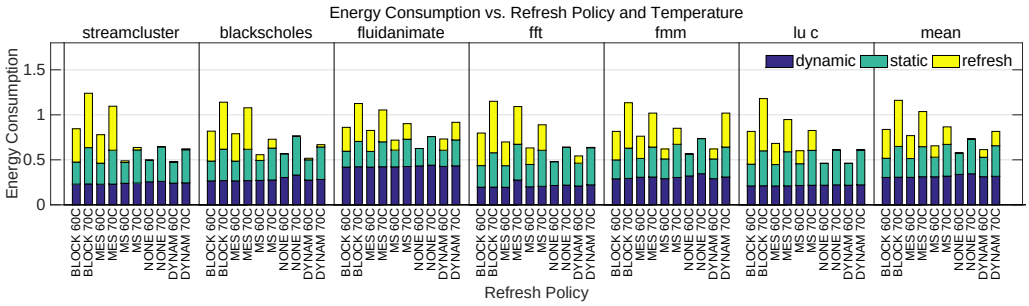


Figure 6.10: Energy consumption of coherent caches (L1+L2) for different refresh policies and temperatures normalised to the SRAM coherent cache for the MESI protocol

Figure 6.10 presents the energy consumption for different temperatures and refresh policies. Results are normalised to the SRAM cache at $60°C$. It can be seen that the refresh energy can increase significantly due to the reduction of the cell retention time. Although Figure 6.10 shows that the overall energy of DRAM cache for BLOK refresh policy is higher than base-

line, it should be noted that the for $70°C$ static power consumption of SRAM cache increase also and that is much higher than DRAM leakage.

The performance loss of the extra refresh actions may be significant when compared to the SRAM baseline as it can be seen on the Figure 6.9. This loss is caused by the reduction in the retention time in higher temperatures. While no refreshing is badly hurt with the smaller retention time, the dynamic scheme is capable of keeping losses below 5% (comparing to the SRAM baseline) even in the hottest scenarios.

In the high-temperature scenarios, refresh actions are more frequent. This benefits our proposal against others since it reduces the number of refresh actions. On the other side, in lower temperature scenarios, the amount of refresh actions is smaller, so the benefits of our proposal over BLOCK refresh diminish. Actually, if the temperature is, unrealistically, constant at $30°C$ (and refresh times increase accordingly), both techniques perform similarly.

6.5.8 Technology Scaling

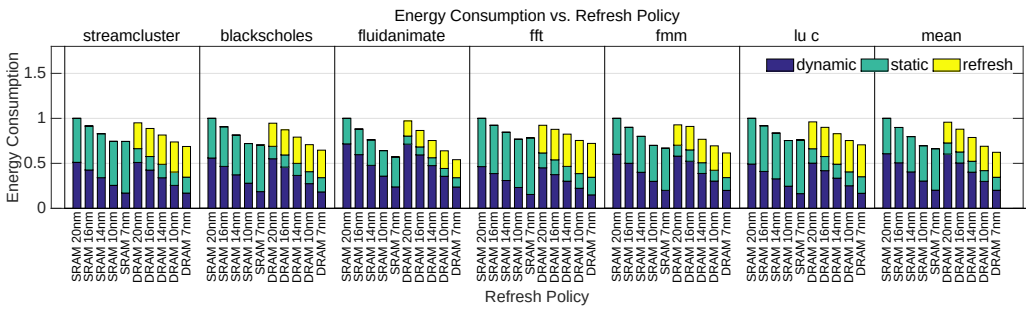


Figure 6.11: The energy consumption of coherent caches (L1+L2) different technologies for the SRAM and DRAM caches, when block refresh is applied

In this section, we analyse the contribution of memory refresh energy on the total memory consumption for the different technologies. For making this plot, we use the ITRS report [1] to obtain the trend for the memory leakage and the dynamic power through technologies. Also, for estimation of the cell retention time we used HSPICE simulation. We assumed the threshold voltage variation through technologies according to [6]. According to our

simulation, a significantly higher retention time can be made for the higher technologies (3.5-4 higher for the 20nm FinFETs than for the 10nm devices) for the same environmental conditions (temperature, voltage).

Figure 6.11 shows that, as the technology shrinks, the dynamic energy is going to continue to reduce. The results are normalised to the 20nm technology node. Figure 6.11 presents the energy consumption for the SRAM and DRAM caches with the BLOCK refresh. At the same time, a further increase in memory leakage is expected. For this plot, we assumed a constant working frequency of 2.5GHz as in our previous results.

On the other hand, due to the reduction in the retention time for the lower technology nodes, we expect a higher contribution of refresh energy. Following the results in Figure 6.11, we can conclude that smart refresh techniques have a smaller impact on the refresh contribution to the overall energy consumption in higher technologies. However, for the future sub 20nm technologies where the retention time is significantly less, our proposal, of reducing refresh energy, has a big impact on the overall memory energy consumption.

6.5.9 Cell Ageing

We used cell access time as a primary metric of the cell ageing. In order to estimate cell ageing we followed the reaction diffusion method explained in the section 2.2.3. The major assumption that we used in our simulation is that device ageing is proportional to the activity factor $f_{AC}(S_p)$ according to the Equation 6.1.

$$\Delta V_{TH} \propto \frac{qN_{IT}(t)}{C_{OX}} \propto f_{AC}(S_p) \times K_{DC} \times t^n \tag{6.1}$$

Here are detailed simulation steps for evaluations.

1. Obtain full system simulation of the baseline system. From the simulation extract the average cache occupation. When a cache line is invalid, it holds "0" in the every cell (as the value will have leaked away). During this period, the PD transistor is not exposed to PBTI. This can be

interpreted as the reduction of S_p in the PD transistor (which ages the most due to the PBTI);

2. From the activity results in step 1) calculate V_{THBTI} shift due to the PBTI based on the Equation 6.1. The results are pre-calculated based on the results presented in [13]. As in [15], we assume a linear dependency of V_{TH} to S_p.

3. Perform HSPICE simulations to calculate the cell read access time with the newly computed threshold voltage V_{THPBTI} in the PD transistor.

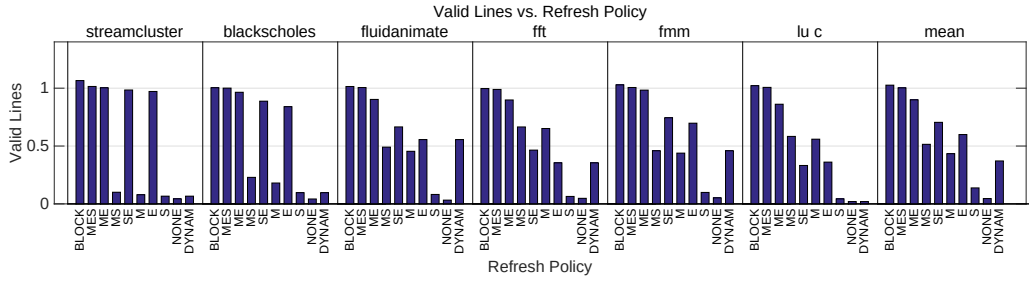


Figure 6.12: Total valid lines for different refresh policies normalised to the SRAM coherent cache for the MESI protocol

Figure 6.12 presents the memory occupation over time for the different refresh policies. In general, the most of the cache is occupied by lines that are in "Modified" or "Exclusive" state and a very small number of lines are in "Shared" state. Consequently, the system execution time increases differently when the different refresh policy is applied, so a trade-off between memory occupation and the system performance should be found.

When cache occupation is used to calculate the ageing (Read Access Time) according to the Equation 6.1, it can be seen that the cell ageing process can be significantly reduced by different refresh policy and benchmark (Figure 6.13). For the policies that occupy the most of the cache memory, we observed an increase in the cell Read Access Time up to 16%. However, when the refresh policies are applied, this number is reduced to just 2% (NONE) or 3%(S) on average. However, as it has been already shown in Figure 6.4, these

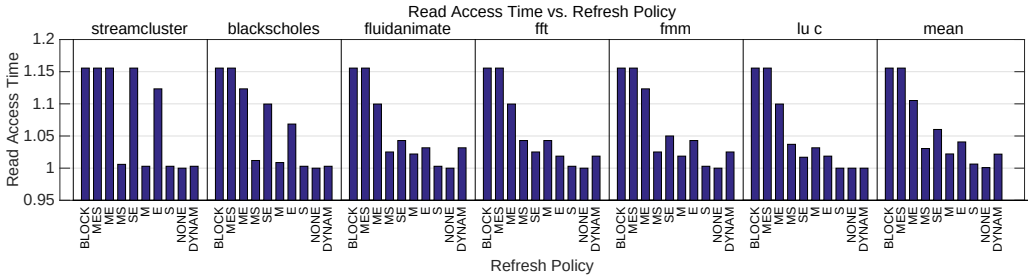


Figure 6.13: Read access time for different refresh policies normalised to the SRAM coherent cache for the MESI protocol

policies produce the largest performance loss. Other refresh policies achieve a trade-off between the performance degradation and the cell ageing. For example, if only lines in the Exclusive state are refreshed, the degradation is reduced by 4x (16% reduced to 4% nominal) with a performance loss of 7%. Ageing reduction goes to 3% for the dynamic approach. Benefits of this proposal are even higher for larger periods of time (5-10 years) because the nominal ageing of the cache is greater.

6.5.10 Retention Time

According to our analysis, the retention time of the cell is not affected significantly by the cell ageing. There are two reasons for that:

- The gate capacitance of the PD transistor (that holds the value of the cell) does not depend on the voltage shift due the ageing.
- The probability of activating the WR transistor of a particular word in the cache (interval when the WL is "1") is very small. Additionally, even for higher probability, the ageing of the WR transistor and V_{TH} increase would manifest as a reduction in the WR transistor leakage. According to that, it is reasonable to expect that the retention time of this cell would increase a little due to the ageing.

This small increase in the retention time of the cell is at the cost of an increase in the cell write access time that would increase slightly, too.

However, these effects are expected to be subtle (low probability of activating WR) and because of that we excluded it from our analysis.

6.5.11 Comparison of the 3T and the 6T cell ageing with respect to the signal probability

The good side of gain cells in terms of ageing is that they are not symmetric. When compared to the 6T SRAM cell that comprises two paired inverters, the gain cell is one sided. As a consequence of the 6T cell symmetry, the minimal ageing of it is achieved when the signal probability is $S_i = 0.5$ (this means that both inverters are equally stressed). Any shift of the signal probability in any direction will cause that one PD transistor (as well as its diagonal PU transistor) to have a higher V_{THBTI} shift which will eventually result in higher ageing (i.e. a noise margin reduction and an increase in the read access time).

On the other side, there is only one PD transistor in a gain cell whose ageing is proportional to the probability of writing "1". In other words, if the probability of writing "0" in the gain cell (3T) is higher than 0.5, the ageing of the PD transistor is going to be smaller than in the 6T SRAM (if the probability of writing "0" is less than 0.5, a simple inverted logic of writing should be applied and the same results results are achieved).

6.5.12 Cache coherency ageing reduction with respect to signal probability

Although the signal probability affects a gain cell ageing less than the 6T cell; it is good to give some presentation of its effects on the cell ageing. Here, we present how our proposal affects cell ageing for two signal probabilities.

Figure 6.14 shows the ageing of the cache when the probability of writing "1" in the cell is 0.25 and 0.10. It can be noted that the ageing of the baseline system drops significantly when the signal probability is smaller. The increase in the read access time is 3% for signal $S_i = 0.25$ for the baseline cache architecture. When the signal probability is $S_i = 0.10$, this

value is even smaller. In the case of such a small degradation, benefits of our proposal are negligible.

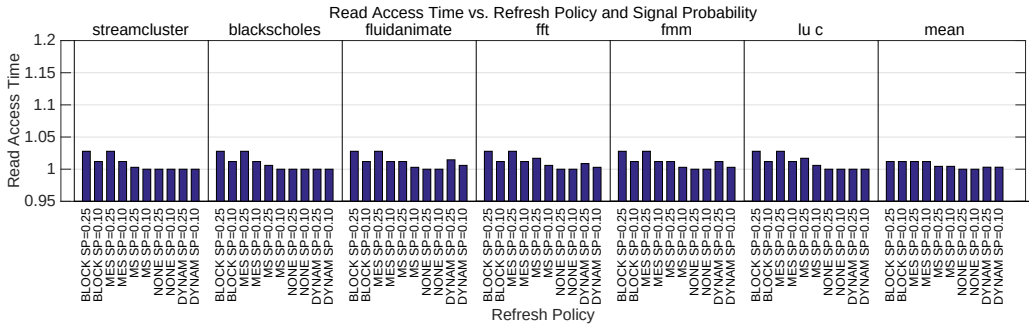


Figure 6.14: Read access time for different refresh policies and signal probabilities normalised for the SRAM coherent cache for the MESI protocol

6.6 Application of coherency based cache refresh on other cache architectures

The results that are presented in the previous sections, clearly state the refresh energy savings with a small performance loss for a particular system configuration. However, considering the vast diversity in the cache architectures it is very hard to find some optimal solution that is reasonably applicable to the every cache configuration. In other words, optimisation that delivers certain advance for one type of cache, does not necessarily have the same effects on some other. In that environment it is hard even to make some fair comparison between different proposals because the most of the optimisations are applicable only for particular configuration and environmental conditions from the great diversity of all possible cases. Because of that we have not made some comparison of this proposals with some work from many papers that appear every year. We, rather, only wanted to prove the hypothesis that cache coherency can be used as an insight for reducing refresh energy under certain conditions. However, it is good to give some general range of application of this proposal as well as to give some directions in the analysis of the idea of using gain cells in a high-speed cache

memory design.

Some rules should be clear. Cache size is very important. For instance, data in larger caches exist for greater period of time, and only applying an optimisation on the caches which size is 2MB (or higher) could lead to higher performance loss than when the same optimisation is applied on 32KB caches if the retention time of a memory cell is the same. The reason for this could be different (e.g. a 2MB cache is greater and slower and it is reasonably to assume that miss penalty is higher). This of course depends of the application too, especially of the application footprint size.

In general, two significant parameters define the range of application of our proposal; cache size and cell retention time. For instance, if the retention time is too small, performance loss will be too high, but the greater amount of refresh energy could be saved. Similarly, for a larger caches more significant amount of energy is needed for refresh.

We have proven in this work that cache coherency can be used for finding trade-off between performance loss and energy savings under certain conditions. However, its application is limited to the small number of cache architectures. If the retention time is big enough, refresh energy will not contribute overall cache memory energy budget, so that kind of optimisation does not have sense. If the retention time is too small (comparing to the cache size) cache memory will lose its purpose since the performance lost will be very high.

Besides these two parameters, a set of other variables have influence, too. The overall contribution of the leakage in the cache energy budget. If the percentage of the leakage in the budget is very small comparing to a gain cell memory, the idea of gain cell memory does not apply.

The leakage strongly depends on the circuit temperature so indirectly temperature plays a significant role in the leakage energy budget as well as in the retention time. Similar conclusions can be said for the technology node used for a cache implementation.

Also, the type of application and its configuration is very important. In this work, we used set of parallel benchmarks that are available to us, but simulate all application that exist or could exist in the future is impossible

to cover in one work.

6.7 Conclusion

This chapter describes how the coherency protocol can be exploited to reduce refresh energy and cache ageing in the dynamic high-speed cache memories. At first, we defined line refresh policies based on the cache coherence state and then we presented how to find the optimal refresh policy that keeps performance high while minimising refresh energy. In addition, it has been investigated how the performance and energy consumption of the system are affected when the memories are exposed to the effects of process and environmental variations. These variations translate to a higher memory leakage, a lower retention time and, eventually, a higher the refresh energy.

In our analysis, we show the significant savings in the refresh energy of coherent caches, up to 37%, and a very small increase in the execution time (up to 5% for MESI coherence protocol) by coherency-based refresh. The energy savings are even higher in the conditions of a higher chip temperature or lower cell retention time as a consequence of the process and environmental variations.

Results also show that the read access time degradation of the memory can be reduced more than 4x when this proposal is applied (3% increase over three years comparing to the baseline degradation of 16%).

However, a bad side of this approach is its range of application. Significant benefits of energy reduction work under certain conditions - chip temperature, cell retention time, technology and cache size, and signal probability when ageing reduction is considered.

It was obvious that this joke was lethal... No one could read it and live ...

Monty Python, The Funniest Joke In The World

7

Conclusions and Future Work

7.1 Summary of Contributions

With continuous technology scaling issues related to the power, and the reliability tends to increase with every generation. The innovations from the every level of the design abstraction are necessary in order to fight the process and environmental variation in order to achieve the low power consumption and satisfying chip yield.

The Introduction of the FinFET technology in the IC design 3 years ago by Intel Corporation has prolonged the Moore's law by solving the problem of process variation caused by Random Discrete Dopants by manufacturing the transistor gate from more than one side. However, variations caused by LER and MGG are still going to pose a significant problem in transistor production, especially for the 10nm technology nodes. This thesis tries to address and investigate the problems that arise in the 10nm FinFET devices when used in the high speed cache design, and to propose some efficient solutions of those issues.

The major contribution of the thesis are:

- Presentation of an original method to simulate process variability when a standardised BSIM-CMG model cards are used in HSPICE Monte Carlo circuit simulation. Additionally, we presented how the Tri-Gate FinFET models can be used for simulating static properties of the circuits that are implemented by Independent Gate FinFETs.

- Characterisation of the classical 6T and 8T SRAM cells as the most sensitive units to the process and environmental variations. We show that although, these circuits can demonstrate high stability in terms of the RSNM and WLNM for the nominal value of the V_{DD} its stability (especially the WLNM) can be significantly compromised when the nominal value of supply voltage is reduced. Some improvement in the WLNM can be achieved by using the IG FinFETs, and the reverse back-gate biasing is applied to the PU transistor, but this goes on the cost of complicated routing.

- Characterisation of the dynamic Gain Cell for the 10nm FinFETs, the comparison with the classical SRAM circuits and proposition of the techniques for increasing retention time for this cell. We have shown that the speed of this cell (RAT) is in the range of the 6T SRAM, and the area is 40% smaller.

- Proposal of one micro-architecture solution that can be used to minimise refresh energy of the high-speed coherent caches when they are implemented by the dynamic 3T gain cells. The technique uses the cache coherence state as a proxy for finding the minimal number of cache lines that should be refreshed and not degrading the performance very much at the same time. Additionally, this solution can be also used for the mitigation of the cell ageing since the lines that are not refreshed are less active and accordingly their lifetime can be extended.

7.2 Future Work

Reliability of the memory and logic circuits designed for the future FinFET technology is the crucial topic that should be discussed researched further. Although some initial investigation of the DRAM-based caches is covered in this thesis, the reliability of complex cache structures is critical, and it can be extended.

A study of the gain cell reliability is shown in Chapter 6 when the cache coherency is used to mitigate cache ageing. However, the development of an efficient method for reliability estimation and solutions to keep memory yield high is still work that can be extended. The major idea, that we plan to do in the future, is to develop an efficient analytical model for calculating memory yield with a superb precision. As the input this tool would use parameters from different levels of design abstraction (process and environmental variability, cell layout parameters, cache micro-architecture configuration...) and it should provide an efficient reliability estimation of such structure in a very short time. Using this type of tool would be highly beneficial because the evaluation of system reliability at early stage is very important and traditional techniques based on the Monte Carlo simulations are time consuming and limited to a certain level of design abstraction. Even the recent proposals that use Importance Sampling (IS) although that can speed up simulation time for a couple of orders of magnitude, they are still slow and their range of application is limited to the particular cell type and cache micro-architecture configurations.

The second thing that we plan to do in the future is the investigation of low power cache memories. Current trends in microprocessor design put circuit power consumption as a critical parameter and accordingly, the devices that work in the near threshold and threshold domain. Characterisation of these devices and circuits demands novel techniques to combat process variability. Reliability is a significant issue in these devices; i.e. the margin for error is significantly reduced due to the small supply voltage. Accordingly, algorithms and circuits have to be modified and adjusted to the environment of the increased number of failures.

7.3 Publications

- *Enhancing 6T SRAM Cell Stability by Back Gate Biasing Technique for 10nm SOI FinFETs under Process and Environmental Variations*, Z. Jakšić, R. Canal, Mixed Design of Integrated Circuits and Systems (MIXDES), 2012 Proceedings of the 19th International Conference;

- *Enhancing 3T DRAMs for SRAM replacement under 10nm tri-gate SOI FinFETs*, Z. Jakšić, R. Canal, Computer Design (ICCD), 2012 IEEE 30th International Conference on;

- *Effects of FinFET Technology Scaling on 3T and 3T1D Cell Performance Under Process and Environmental Variations*, Z. Jakšić, R. Canal, Workshop on Resilient Architectures, Held in Conjunction with 45th International Symposium on Microarchitecture (MICRO), 2012

- *Comparison of SRAM Cells for 10-nm SOI FinFETs Under Process and Environmental Variations*, Z. Jakšić, R. Canal, Electron Devices, IEEE Transactions on, January, 2013;

- *DRAM-based Coherent Caches and How to Take Advantage of the Coherence Protocol to Reduce the Refresh Energy*, Z. Jakšić, R. Canal, Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014

- *Coherency Based Refresh and Degradation Adjustment in FinFET-based DRAM Caches*, Z. Jakšić, R. Canal, Design, Submitted to the ACM Transactions on Architecture and Code Optimization (TACO)

Bibliography

[1] "http://www.itrs.net/," *International Technology Roadmap for Semiconductors*, 2013. 2, 44, 80, 81, 86

[2] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," *Proceedings of IEEE*, vol. 98, no. 10, pp. 1718–1751, October 2010. 3, 17

[3] S. Ganapathy, R. Canal, A. Gonzalez, and A. Rubio, "Circuit propagation delay estimation through multivariate regression-based modeling under spatio-temporal variability," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2010*. 3, 28

[4] S. Tawfik, Z. Liu, and V. Kursun, "Independent-gate and tied-gate finfet sram circuits:design guidelines for reduced area and enhanced stability," in *Microelectronics, ICM Internatonal Conference on (ICM), 2007*. 3, 24, 25, 33, 34, 39, 54

[5] M. Fan, Y. Wu, V. Hu, P. Su, and C. Chuang, "Comparison of 4t and 6t finfet sram cells for subthreshold operation considering variability, a model-based approach," *Electron Devices, IEEE Transactions on*, vol. 58, no. 3, pp. 609–616, March 2011. 3

[6] X. Wang, A. Brown, B. Cheng, and A. Asenov, "Statistical variability and reliability in nanoscale finfets," in *Electron Devices Meeting (IEDM), IEEE International , 2011*. xvii, 3, 10, 11, 24, 26, 28, 51, 86

[7] A. Asenov, A. Brown, J. Davies, S. Kaya, and G. Slavcheva, "Simulation of intrinsic parameter fluctuations in decananometer and nanometer-

scale mosfets," *IEEE transactions on electron devices*, vol. 50, no. 9, pp. 1837–1852, September 2003. 3, 26

[8] E. Baravelli, M. Jurczak, N. Speciale, K. Meyer, and A. Dixit, "Impact of ler and random dopant fluctuations on finfet matching performance," *IEEE transactions on nanotechnology*, vol. 7, no. 3, pp. 291–298, March 2008. 3, 4, 31

[9] S. Ganapathy, *Reliability In The Face of Variability in Nanometer Embedded Memories.* Universitat Politecnica de Catalunya, 2014. 8, 9, 16

[10] M. H. Abu-Rahma and M. Anis, *Nanometer Variation-Tolerant SRAM Statistical Design for Yield.* Springer, 2013. 9

[11] D. Lu, A. Niknejad, C. Hu, and C. Lin, "Compact modeling in variations of finfet sram cells," *Design and Test of Computers, IEEE*, vol. 27, no. 2, pp. 45–50, February 2010. 9, 26, 33, 36, 54

[12] S. Bhunia and S. Mukhopadhyay, *Low-Power Variation-Tolerant Design in Nanometer Silicon.* Springer, 2011. 10, 11, 12

[13] S. Khan, I. Agbo, S. Hamdioui, H. Kukner, B. Kaczer, P. Raghavan, and F. Catthoor, "Bias temperature instability analysis of finfet based sram cells," March 2014. 13, 14, 66, 88

[14] S. Kumar, C. Kim, and S. Sapatnekar, "An analytical model for negative bias temperature instability," in *Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on*, Nov 2006, pp. 493–496. 14

[15] K. Kang, S. Gangwal, S. P. Park, and K. Roy, "Nbti induced performance degradation in logic and memory circuits: how effectively can we approach a reliability solution?" in *Design Automation Conference, 2008. ASP-DAC 2008. Asia and South Pacific*, March 2008, pp. 726–731. 14, 88

[16] J. A. Blome, S. Gupta, S. Feng, and S. Mahlke, "Cost-efficient soft error protection for embedded microprocessors," in *Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, 2006. 15

[17] V. Degalahal, L. Li, V. Narayanan, M. Kandemir, and M. Irwin, "Soft errors issues in low-power caches," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 10, pp. 1157–1166, October 2005. 15

[18] A. Bonnoit, , and L. Pileggi, "Reducing variability in chip-multiprocessors with adaptive body biasing," in *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*. 16, 26

[19] Y. Yasuda, Y. Akiyama, Y. Yamagata, Y. Goto, and K. Imai, "Design methodology of body-biasing scheme for low power system lsi with multi- vth transistors," *Electron Devices, IEEE transactions on*, vol. 54, no. 11, pp. 2946–2952, November 2007. 16, 26

[20] J. Tschanz, S. Narendra, R. Nair, and V. De, "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 5, pp. 826–829, May 2002. 16

[21] ——, "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," in *VLSI Circuits Digest of Technical Papers, 2002, Symposium on*. 16

[22] S. Ganapathy, R. Canal, A. Gonzalez, and A. Rubio, "Dynamic fine-grain body biasing of caches with latency and leakage 3t1d-based monitors," in *Computer Design (ICCD), 2011 IEEE 29th International Conference on*. 16

[23] D. Ernst, N. S. Kim, S. Das, S. Pant, T. Pham, R. Rao, C. Ziesler, D. Blaauw, T. Austin, and T. Mudge, "Razor: A low-power pipeline

based on circuit-level timing speculation," in *Microarchitecture, 2003. MICRO-36. 2003 41st IEEE/ACM International Symposium on*. 16

[24] S. Ghosh, S. Mukhopadhyay, K. Kim, and K. Roy, "Self-calibration technique for reduction of hold failures in low-power nano-scaled sram," in *Design Automation Conference, 2006 43rd ACM/IEEE*. 17

[25] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of sram array for yield enhancement in nanoscaled cmos," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 24, no. 12, pp. 1859–1880, Dec 2005. 17

[26] H. Pilo, C. Barwin, G. Braceras, C. Browning, S. Lamphier, and F. Towler, "An sram design in 65-nm technology node featuring read and write-assist circuits to expand operating voltage," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 4, pp. 813–819, April 2007. 18

[27] K. Nii, M. Yabuuchi, Y. Tsukamoto, S. Ohbayashi, S. Imaoka, H. Makino, Y. Yamagami, S. Ishikura, T. Terano, T. Oashi, K. Hashimoto, A. Sebe, S. Okazaki, K. Satomi, H. Akamatsu, and H. Shinohara, "A 45-nm bulk cmos embedded sram with improved immunity against process and temperature variations," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 180–191, January 2008. 18

[28] N. Weste and D. Haris, *CMOS VLSI Design*. Pearson Addison Wesley, 2005. 19, 32, 44, 45, 48

[29] A. Agarwal, B. Paul, H. Mahmoodi, A. Datta, and K. Roy, "A process-tolerant cache architecture for improved yield in nanoscale technologies," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 27–38, January 2005. 19, 20

[30] M. Qazi, M. Sinangil, and A. Chandrakasan, "Challenges and directions for low-voltage sram," *Design and Test of Computers, IEEE*, vol. 28, no. 1, pp. 32–43, January 2011. 19, 34

[31] K. Takeda, Y. Hagihara, Y. Aimoto, M. Nomura, Y. Nakazawa, T. Ishii, and H. Kobatake, "A read-static-noise-margin-free sram cell for low-vdd and high-speed applications," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 1, pp. 113–121, January 2006. 19

[32] B. Calhoun and A. Chandrakasan, "A 256kb 65nm sub-threshold sram design for ultra-low-voltage operation," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 3, pp. 680–688, March 2007. 19

[33] I. J. Chang, J.-J. Kim, S. P. Park, and K. Roy, "A 32 kb 10t sub-threshold sram array with bit-interleaving and differential read scheme in 90 nm cmos," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 2, pp. 650–658, February 2009. 19

[34] L. Chang, R. Montoye, Y. Nakamura, K. Batson, R. Eickemeyer, R. Dennard, W. Haensch, and D. Jamsek, "An 8t-sram for variability tolerance and low-voltage operation in high-performance caches," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 4, pp. 956–963,, April 2008. 19

[35] L. Chang, D. Fried, J. Hergenrother, J. Sleight, R. Dennard, R. Montoye, L. Sekaric, S. McNab, A. Topol, C. Adams, K. Guarini, and W. Haensch, "Stable sram cell design for the 32 nm node and beyond," in *VLSI Technology, 2005. Digest of Technical Papers. 2005 Symposium on.* 19

[36] W. Luk, J. Cai, R. Dennard, M. Immediato, and S. Kosonocky, "A 3-transistor dram cell with gated diode for enhanced speed and retention time," in *VLSI Circuits, 2006. Digest of Technical Papers. 2006 Symposium on.* 19, 49

[37] W. Luk and R. Dennard, "2t1d memory cell with voltage gain," in *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on.* 19, 49

[38] ——, "A novel dynamic memory cell with internal voltage gain," *Solid-State Circuits, IEEE journal of*, vol. 40, no. 4, pp. 884–894, April 2005. 19, 48, 49

[39] D. Patterson, P. Garrison, M. Hill, D. Lioupis, C. Nyberg, T. Sippel, and K. V. Dyke, "Architecture of a vlsi instruction cache for a risc," in *International Symposium Computer Architecture (ISCA), 2003, Procedings of*. 20

[40] D. C. Bossen, J. M. Tendler, and K. Reick, "Power4 system design for high reliability," *IEEE Micro*, vol. 22, no. 2, pp. 16–24, March/April 2002. 20

[41] M. Manoochehri, M. Annavaram, and M. Dubois, "Extremely low cost error protection with correctable parity protected cache," *Computers, IEEE Transactions on*, vol. 63, no. 10, pp. 2431–2444, Oct 2014. 21

[42] D. J. Sorin, *Fault Tolerant Computer Architectures*. Morgan & Claypool Publishers, 2009, 2009. 21

[43] I. Corporation, in *Intel Pentium 4 Processor on 90nm Process Datasheet*, April, 2004. 21

[44] S. Microsystems, in *UltraSPARC IV Processor Architecture Overview. Sun Microsystems Technical Whitepaper*, February, 2004. 21

[45] J. Kim, N. Hardavellas, K. Mai, B. Falsafi, and J. Hoe, "Multi-bit error tolerant caches using two-dimensional error coding," in *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, 2007. 21

[46] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: simple techniques for reducing leakage power," in *Computer Architecture, 2002. Proceedings. 29th Annual International Symposium on*, 2002. 21

[47] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache decay: exploiting generational behavior to reduce cache leakage power," in *Computer Architecture, 2001. Proceedings. 28th Annual International Symposium on*, 2001. 21

[48] Z. Jaksic and R. Canal, "Enhancing 6t sram cell stability by back gate biasing techniques for 10nm soi finfets under process and environmental variations," in *International Conference of Mixed Design of Integrated Circuits and Systems (MIXDES), 2012*. 24, 32

[49] ——, "Comparison of sram cells for 10-nm soi finfets under process and environmental variations," *Electron Devices, IEEE Transactions on*, vol. 60, no. 1, pp. 49–55, January 2013. 24, 32, 51, 54

[50] J. P. Colinge, *FinFETs and Other Multigate Transistors*. Springer, 2008. 24, 26

[51] "Synopsys: Hspice reference manual," September 2011. 26

[52] "Garand simulator: online: www.goldstandardsimulations.com." 27

[53] A. Asenov, S. Kaya, and A. Brown, "Intrinsic parameter fluctuations in decananometer mosfets introduced by gate line edge roughness," *Electron Devices, IEEE Transactions on*, vol. 50, no. 5, pp. 1254–1260, May 2003. 27

[54] A. R. Brown, N. M. Idris, J. R. Watling, and A. Asenov, "Impact of metal gate granularity on threshold voltage variability: A full-scale three-dimensional statistical simulation study," *IEEE Electron Device Letters*, vol. 31, no. 11, pp. 1199–1201, November 2010. 27

[55] X. Wang, A. Brown, N. Idris, S. Markov, G. Roy, and A. Asenov, "Statistical threshold-voltage variability in scaled decananometer bulk hkmg mosfets: A full-scale 3-d simulation scaling study," *Electron Devices, IEEE Transactions on*, vol. 58, no. 8, pp. 2293–2301, August 2011. 27

[56] A. R. Brown, J. R. Watling, and A. Asenov, "Intrinsic parameter fluctuations due to random grain orientations in high-k gate stacks," *Journal of Computational Electronics*, vol. 5, no. 4, pp. 333–336, April 2006. 28

[57] A. Agawal, D. Blauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations," in *Computer Aided Design, International Conference on (ICCAD), 2003.* 28

[58] Z. Liu, S. Tawfik, and V. Kursun, "Statistical data stability and leakage evaluation of finfet sram cells with dynamic threshold voltage tuning under process parameter fluctuations," in *Quality Electronic Design, 9th International Symposium on (ISQED) , 2008.* 33, 34, 37

[59] S. Tawfik and V. Kursun, "Work-function engineering for reduced power and higher integration density an alternative to sizing for stability in finfet memory circuits," in *Circuits and Systems, IEEE International Symposium on (ISCAS) , 2008.* 34

[60] S. O'uchi, K. Endo, M. Masahara, K. Sakamoto, Y. Liu, T. Matsukawa, T. Sekigawa, H. Koike, and E. Suzuki, "Flex-pass-gate sram for static noise margin enhancement using finfet-based technology," *Solid-State Electronics*, vol. 52, no. 7, pp. 169–1702, July 2008. 35

[61] M. Fan, Y. Wu, V. Hu, P. Su, and C. Chuang, "Investigation of cell stability and write ability of finfet subthreshold sram using analytical snm model," *Electron Devices, IEEE transactions on*, vol. 57, no. 6, pp. 1357–1381, June 2010. 35, 37, 38

[62] ——, "Investigation of stability and ac performance of sub-threshold finfet sram," in *VLSI Technology Systems and Applications, International Symposium on (VLSI-TSA) , 2011.* 35, 37, 38

[63] A. Carlson, Z. Guo, S. Balasubramanian, R. Zlatanovici, T. Liu, and B. Nikolic, "Sram read/write margin enhancements using finfets," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 8, no. 6, pp. 887–900, June 2010. 35

[64] C. Shin, Y. Tsukamoto, X. Sun, and T. Liu, "Full 3d simulation of 6t-sram cells for the 22nm node," in *Simulation of Semiconductor Processes and Devices, International Conference on (SISPAD), 2009.* 35

[65] A. Sachid, R. Francis, M. Baghini, D. Sharma, and K. Bach, "Sub-20 nm gate length finfet design: Can high-k spacers make a difference?" in *Electron Devices Meeting, 2008. IEDM 2008. IEEE International*. 35, 37, 38

[66] J. G. Delgado-Frias, Z. Zhang, and M. Turi, "Low power sram cell design for finfet and cntfet technologies," in *Green Computing Conference, 2010 International*. 35, 37, 43

[67] M. Turi and J. Delgado-Frias, "Performance-power tradeoffs of 8t finfet sram cells," in *Circuits and Systems, IEEE International Midwest Symposium on (MWSCAS), 2011*. 35, 37, 43

[68] S. Gupta, S. Park, and K. Roy, "Tri-mode independent-gate finfets for dynamic voltage/frequency scalable 6t srams," *Electron Devices, IEEE Transactions on*, vol. 58, no. 11, pp. 3837–3846, November 2011. 35, 43

[69] P. Stolk, H. Tuinhout, R. Duffy, E. Augendre, L. P. Bellefroid, M. J. B. Bolt, J. Croon, C. J. J. Dachs, F. R. J. Huisman, A. J. Moonen, Y. Ponomarev, R. F. M. Roes, M. Da Rold, E. Seevinck, K. N. Sreerambhatla, R. Surdeanu, R. M. D. A. Velghe, M. Vertregt, M. N. Webster, N. K. J. Van Winkelhoff, and A. T. A. Zegers-Van Duijnhoven, "Cmos device optimization for mixed-signal technologies," in *Electron Devices Meeting, 2001. IEDM '01. Technical Digest. International*, 2001. 38

[70] B. Cheng, A. Brown, X. Wang, and A. Asenov, "Statistical variability study of a 10nm gate length soi finfet device," in *Silicon Nanoelectronics Workshop (SNW), 2012 IEEE*, 2012. 38

[71] S. Gupta, J. Kulkarni, and K. Roy, "Tri-mode independent gate finfet-based sram with pass-gate feedback: Technology-circuit co-design for enhanced cell stability," *Electron Devices, IEEE Transactions on*, vol. 60, no. 11, pp. 3696–3704, Nov 2013. 44

[72] Z. Jaksic and R. Canal, "Enhancing 3t drams for sram replacement under 10nm tri-gate soi finfets," in *Computer Design (ICCD), 2012 IEEE 30th International Conference on*. 48, 52, 63, 64, 77, 78, 83

[73] A. Bhoj and N. Jha, "Pragmatic design of gated-diode finfet drams," in *Computer Design, 2009. ICCD 2009. IEEE International Conference on*. 49

[74] K. Lovin, B. Lee, X. Liang, D. Brooks, and G. Wei, "Empirical performance models for 3t1d memories," in *Computer Design, 2009. ICCD 2009. IEEE International Conference on*. 50

[75] X. Liang, R. Canal, G. Wei, and D. Brooks, "Process variation tolerant 3t1d-based cache architectures," in *Proceedings of the International Symposium on Microarchitecture (MICRO), 2007*. 50, 57, 62, 63, 78

[76] Z. Hassan, N. Allec, L. Shang, R. Dick, V. Venkatraman, and R. Yang, "Multiscale thermal analysis for nanometer-scale integrated circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 6, pp. 860–873, June 2009. 51

[77] M.-T. Chang, P. Rosenfeld, S.-L. Lu, and B. Jacob, "Technology comparison for large last-level caches (l3cs): Low-leakage sram, low write-energy stt-ram, and refresh-optimized edram," in *International Symposium on High Performance Computer Architecture (HPCA), 2013*. 62, 63, 66, 81

[78] "Intel ivy bridge cmps produc series: online: http://ark.intel.com/products, 2015." 62

[79] "Nvidia tegra 4 family cpu architecture whitepaper," *www.nvidia.com, 2015*. 62

[80] Z. Jaksic and R. Canal, "Dram-based coherent caches and how to take advantage of the coherence protocol to reduce the refresh energy," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*. 63, 72

[81] M. Ghosh and H.-H. Lee, "Smart refresh: An enhanced memory controller design for reducing energy in conventional and 3d die-stacked drams," in *Microarchitecture, 2007. MICRO 2007. 40th Annual IEEE/ACM International Symposium on.* 63, 64

[82] X. Liang, R. Canal, G.-Y. Wei, and D. Brooks, "Replacing 6t srams with 3t1d drams in the l1 data cache to combat process variability," *Micro, IEEE*, vol. 28, no. 1, pp. 60–68, 2008. 63

[83] M.-T. Chang, P. Rosenfeld, S.-L. Lu, and B. Jacob, "Refresh matters: Energy and performance analysis of large last-level cache built with gain cell embedded dram, http://hdl.handle.net/1903/13296, technical report," 2013. 63

[84] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, "Raidr: Retention-aware intelligent dram refresh," in *Computer Architecture (ISCA), 2012, International Symposium on.* 64

[85] S. Liu, K. Pattabiraman, T. Moscibroda, and B. G. Zorn, "Flikker: Saving dram refresh-power through critical data partitioning," in *Proceedings on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2011.* 64

[86] J. Stuecheli, D. Kaseridis, H. Hunter, and L. John, "Elastic refresh: Techniques to mitigate refresh penalties in high density memory," in *Microarchitecture, 2010. MICRO 2010. 43th Annual IEEE/ACM International Symposium on.* 64

[87] P. Nair, C.-C. Chou, and M. K. Qureshi, "A case for refresh pausing in dram memory systems," in *High Performance Computer Architecture (HPCA), 2013, IEEE International Symposium on.* 64

[88] J. Mukundan, H. Hunter, K. hyoun Kim, J. Stuecheli, and J. F. Martinez, "Understanding and mitigating refresh overheads in high-density ddr4 dram systems," in *International Symposium on Computer Architecture (ISCA), 2013.* 64

[89] A. Agrawal, P. Jain, A. Ansari, and J. Torrellas, "Refrint: Intelligent refresh to minimize power in on-chip multiprocessor cache hierarchies," in *High Performance Computer Architecture (HPCA), IEEE International Symposium on, 2013.* 64

[90] A. Agrawal, A. Ansari, and J. Torrellas, "Mosaic: Exploiting the spatial locality of process variation to reduce refresh energy in on-chip edram modules," in *High Performance Computer Architecture (HPCA), 2014, IEEE International Symposium on.* 64

[91] J. Abella, X. Vera, and A. Gonzalez, "Penelope: The nbti-aware processor," in *Microarchitecture, 2007. MICRO. 40th Annual IEEE/ACM International Symposium on*, Dec 2007. 65

[92] S. Kumar, C. Kim, and S. Sapatnekar, "Impact of nbti on sram read stability and design for reliability," in *Quality Electronic Design, 2006. ISQED '06. 7th International Symposium on*, March 2006, pp. 6 pp.–218. 65

[93] J. Sun, A. Kodi, A. Louri, and J. Wang, "Nbti aware workload balancing in multi-core systems," in *Quality of Electronic Design, 2009. ISQED 2009. Quality Electronic Design*, March 2009, pp. 833–838. 65

[94] A. Rahimi, L. Benini, and R. Gupta, "Aging-aware compiler-directed vliw assignment for gpgpu architectures," in *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, May 2013, pp. 1–6. 65

[95] M. Basoglu, M. Orshansky, and M. Erez, "Nbti-aware dvfs: A new approach to saving energy and increasing processor lifetime," in *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*, Aug 2010, pp. 253–258. 66

[96] L. Zhang and R. Dick, "Scheduled voltage scaling for increasing lifetime in the presence of nbti," in *Design Automation Conference, 2009. ASP-DAC 2009. Asia and South Pacific*, Jan 2009, pp. 492–497. 66

[97] J. H. Papamarcos, M. S.; Patel, "A low-overhead coherence solution for multiprocessors with private cache memories," in *Computer Architecture (ISCA), 1984, International Symposium on*, 1984. 67

[98] M. Monchiero, R. Canal, and A. Gonzalez, "Using coherence information and decay techniques to optimize l2 cache leakage in cmps," in *Parallel Processing, 2009. ICPP '09. International Conference on.* 67

[99] A. Patel and K. Ghose, "Energy-efficient mesi cache coherence with pro-active snoop filtering for multicore microprocessors," in *Low Power Electronics and Design (ISLPED), 2008, ACM/IEEE International Symposium on.* 67, 76

[100] X. Qin and P. Mishra, "Automated generation of directed tests for transition coverage in cache coherence protocols," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2012.* 67

[101] T. Suh, D. Kim, and H.-H. Lee, "Cache coherence support for non-shared bus architecture on heterogeneous mpsocs," in *Design Automation Conference, 2005. Proceedings. 42nd.* 67

[102] in *AMD64 Architecture Programmer's Manual Vol 2 System Programming*, September 2012. 70

[103] "http://marss86.org/ marss86/index.php/home," *MARSSx86 Home Page*, 2014. 76, 78, 80

[104] A. Patel, F. Afram, S. Chen, and K. Ghose, "Marssx86: A full system simulator for x86 cpus," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE.* 76

[105] C. Bienia, S. Kumar, and K. Li, "Parsec vs. splash-2: A quantitative comparison of two multithreaded benchmark suites on chip-multiprocessors," in *IEEE International Symposium on Workload Characterization (IISWC), 2008.* 78

[106] "http://www.capsl.udel.edu/splash/," *SPLASH-2 Benchmark Suite, Home Page*, 2013. 78

[107] "http://parsec.cs.princeton.edu/," *PARSEC Benchmark Suite, Home Page*, 2013. 78

[108] I. Choi, M. Zhao, X. Yang, and D. Yeung, "Experience with improving distributed shared cache performance on tilera's tile processor," *Computer Architecture Letters*, vol. 10, no. 2, pp. 45–48, 2011. 78