

Multilayer Modeling and Design of Energy Managed Microsystems

By

Houman Zarrabi

A Thesis

In the Department of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

April 2011

© Houman Zarrabi, 2011

**CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: **Houman Zarrabi**

Entitled: **Multilayer Modeling and Design of Energy Managed Microsystems**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Dr. Peter Grogono _____ Chair

_____ Dr. Majid Ahmadi _____ External to University

_____ Dr. Rajagopalan Jayakumar _____ External to Program

_____ Dr. Glenn Cowan _____ Examiner

_____ Dr. M. Zahangir Kabir _____ Examiner

_____ Dr. A. J. Al-Khalili _____ Co-Supervisor

_____ Dr. Yvon Savaria _____ Co-Supervisor

Approved by: _____
Chair of the Department of Electrical and Computer Engineering

_____ _____
Dean, Faculty of Engineering and Computer Science

Abstract

Multilayer Modeling and Design of Energy Managed Microsystems¹

Houman Zarrabi, Ph. D.

Concordia University, 2011

Aggressive energy reduction is one of the key technological challenges that all segments of the semiconductor industry have encountered in the past few years. In addition, the notion of environmental awareness and designing “green” products is yet another major driver for ultra low energy design of electronic systems.

Energy management is one of the unique solutions that can address the simultaneous requirements of high-performance, (ultra) low energy and greenness in many classes of computing systems; including high-performance, embedded and wireless.

These considerations motivate the focus of this dissertation on the energy efficiency improvement of Energy Managed Microsystems (EMM or EM²). The aim is to maximize the energy efficiency and/or the operational lifetime of these systems. In this thesis we propose solutions that are applicable to many classes of computing systems including high-performance and mobile computing systems. These solutions contribute to make such technologies “greener”. The proposed solutions are multilayer, since they belong to, and may be applicable to, multiple design abstraction layers. The proposed solutions are

¹ In this dissertation, by Microsystems, we refer to microelectronic systems.

orthogonal to each other, and if deployed simultaneously in a vertical system integration approach, when possible, the net benefit may be as large as the multiplication of the individual benefits.

At high-level, this thesis initially focuses on the modeling and design of interconnections for EM². For this purpose, a design flow has been proposed for interconnections in EM². This flow allows designing interconnects with minimum energy requirements that meet all the considered performance objectives, in all specified system operating states.

Later, models for energy performance estimation of EM² are proposed. By energy performance, we refer to the improvements of energy savings of the computing platforms, obtained when some enhancements are applied to those platforms. These models are based on the components of the application profile. The adopted method is inspired by Amdahl's law, which is driven by the fact that 'energy' is 'additive', as 'time' is 'additive'. These models can be used for the design space exploration of EM². The proposed models are high-level and therefore they are easy to use and show fair accuracy, 9.1% error on average, when compared to the results of the implemented benchmarks.

Finally, models to estimate energy consumption of EM² according to their "activity" are proposed. By "activity" we mean the rate at which EM² perform a set of predefined application functions. Good estimations of energy requirements are very useful when designing and managing the EM² activity, in order to extend their battery lifetime. The study of the proposed models on some Wireless Sensor Network (WSN) application benchmark confirms a fair accuracy for the energy estimation models, 3% error on average on the considered benchmarks.

Acknowledgments

It was not foreseeable for me to accomplish my doctorate without significant contributions and support from my mentors, my family and my true friends.

First, I would like to express my gratitude to Professor A. J. Al-Khalili. His true understanding and endless support were the main keys and the best motivations for my progress and achievements. He opened the doors for me and let me grow. He always supported me mentally and treated me like his son. It is not easy to express my experience and feelings in a few words but I acknowledge that he is one of the few angels that happened to me and let good things happen to me.

Secondly, I would like to thank Professor Yvon Savaria, for his invaluable and honest guidance and advices throughout this period. His constructive criticisms enhanced my research quality to a great extent. His technical insights along with his positive and supporting attitude, made a perfect advisor for me and I am privileged to be supervised by this distinguished gentleman.

I wish to offer my sincere love to my family, who always supported me with love. I owe them all my life and I wish I have been able to make them happy by this achievement.

Finally, I would like to thank my true friends who were beside me, shared good and memorable time with me during this journey and helped me overcome my difficulties.

Dedication

I dedicate this work to:

- My Mother, who is the symbol of care and love to me,
- My Father, who is the symbol of sacrifice to me,
- My Sister, my nephews and my niece, who are tolerant and have great self-esteem,
- My Mentors, who always grant me serenity and make me stronger,
- My true friends, who care about me, and,
- The People who left us, but we still live with their memories.

Contents

List of Figures	x
List of Tables	xii
List of Acronyms	xiii
Chapter 1 Introduction.....	1
1.1 Introduction	1
1.2 Thesis Motivations	5
1.3 Research problems, solutions and contributions	5
1.3.1 Design exploration of EM ² based on interconnection	5
1.3.2 Design exploration of EM ² based on application	7
1.3.3 Design exploration of EM ² based on activity	8
1.4 Thesis organizations	9
Chapter 2 Energy Reduction Techniques in Various Design Abstraction Layers	10
2.1 Overview	10
2.2 Energy and power basics.....	10
2.3 Minimum Energy Point (MEP).....	12
2.4 Related energy reduction techniques.....	14
2.4.1 Circuit/System level techniques.....	14
2.4.2 System/Application level techniques.....	19
Chapter 3 Design Exploration of Energy Managed Microsystems Based on Interconnection	25
3.1 Chapter overview	25
3.2 Modeling interconnects in EM ²	26
3.2.1 The interconnect performance models.....	26
3.2.2 The impact of technology parameters on the performance models	30
3.2.3 The implications of the performance models.....	35
3.3 Designing interconnects in EM ²	35
3.3.1 Designing interconnects with latency objectives	36
3.3.2 Designing interconnects with frequency objectives	37
3.3.3 Designing interconnects with energy objectives.....	39
3.3.4 Designing interconnects with area objectives.....	39

3.4	Managing interconnects in EM ²	40
3.4.1	The DVS design metrics	40
3.4.2	A compact DVS model using the design metrics	41
3.4.3	Scaling limit for error-free system operation.....	41
3.4.4	Selecting supply voltages.....	42
3.4.5	The interpolation method.....	43
3.5	The flow	46
3.6	Case study: design and management of integrated multi-cycle buses in a power-managed system	47
3.7	Chapter conclusions	57
Chapter 4 Design Exploration of Energy Managed Microsystems Based on Application 58		
4.1	Chapter overview	58
4.2	Foundations	59
4.2.1	Amdahl's Law.....	59
4.2.2	Elements of DVS	61
4.3	System power models.....	63
4.3.1	System models	64
4.3.2	System power models	65
4.4	System energy performance models	67
4.4.1	Platforms with power gating.....	68
4.4.2	Platforms without power gating.....	69
4.5	Energy performance models in systems subject to DVS	71
4.5.1	Platforms with power gating.....	71
4.5.2	Platforms without power gating.....	75
4.6	Validation of the performance models	77
4.6.1	The ASIP platform technology and its design space exploration for energy performance	77
4.6.2	Case studies and implementation results	80
4.6.3	Comparing results	83
4.7	Chapter conclusions	87
Chapter 5 Design Exploration of Energy Managed Microsystems Based on Activity 88		
5.1	Chapter overview	88
5.2	Energy estimation of embedded systems based on their activity.....	89

5.2.1	A generic application-driven energy performance model based on the notion of system activity	89
5.3	Energy performance model for generic ZigBee [®] application platforms based on the notion of activity	91
5.4	Experimental results	93
5.5	Chapter conclusions	98
Chapter 6	Conclusions	100
6.1	Summary	100
6.2	Review of the thesis contributions	101
6.3	Recommendations for future work	102
6.3.1	Future works in the domain of EM ² interconnection realization	103
6.3.2	Future works in the domain of application-level EM ² realization	103
Appendix I	105
Bibliography	106

List of Figures

Figure 1-1: The power density trend of some well-known high-performance processors [2] [3].....	1
Figure 1-2: Mobile electronics complexity vs. battery capacity trends [4].	2
Figure 1-3: The benefit of designing green information processing systems [5].....	3
Figure 1-4: Some major solutions to green information processing systems [5].....	4
Figure 2-1: The minimum energy point trend in DSM [10].	13
Figure 2-2: A model for a power-managed system performing simultaneous DPM and DVS (Inspired from [17]).	15
Figure 3-1: A generic decoupled interconnect model in Microsystems.	26
Figure 3-2: Extracting the approximate value for velocity saturation index α that best matches HSPICE transient simulation results. The delays are normalized by their respective nominal values at $V_{dd}=1$	32
Figure 3-3: Average error when the performance is estimated with different values of α , assumed to hold over the full operating range 0.4-1V, normalized with respect to $\alpha=2$	34
Figure 3-4: The driver delay portion, obtained based on the driver size (h) and the global interconnect length in 45-nm ($k=1, \alpha=2$).....	35
Figure 3-5: Sub-/systems delay subject to voltage scaling, based on the driver delay portion ($\tau d - r$) and the supply-voltage, ($k=1, \alpha=2$).....	36
Figure 3-6: Quantifying FVi for a large spectrum of designs for $\alpha=1$ and $\alpha=2$ in 45nm technology.	44
Figure 3-7: The interconnect design and management flow.....	45
Figure 3-8: The design space exploration to meet the latency objectives in (a) Bus #1, (b) Bus #2; to meet the operating frequency objectives in (c) Bus #1, (d) Bus #2.....	49

Figure 3-9: The required ϑ in the design space exploration of (a) Bus #1 and (b) Bus #2. The design spaces of (c) Bus #1 and (d) Bus #2.	51
Figure 4-1: Modeling computing platforms based on their “resolute” and “enhanced” sets of processing elements.	64
Figure 4-2: Power models of platforms: (a) with power gating, (b) without power gating.	65
Figure 4-3: An application running on platforms: (a) the original case, (b) enhanced with power gating, (c) enhanced without power gating, (d) enhanced with power gating after DVS, (e) enhanced without power gating after DVS.	67
Figure 4-4: Sweeping the architectural dependency (d) in the three case studies, Error < 0 implies overestimating the energy performance.	84
Figure 5-1: Power/Energy profile of an application, including “fixed” and “tunable” segments, running on an embedded system.	89
Figure 5-2: Pseudo code of a generic control monitoring application implemented over TI Z-Stack.	92
Figure 5-3: Experimental characterization of power/energy profiles of a ZigBee® WSN platform during (a) sensing, and, data transmission, (b) data reception.	94
Figure 5-4: Charactering WSN application platform in four different operating states (a) $\alpha s = 0$, (b) $\alpha s = 1$, (c) $\alpha s = 2$, (d) $\alpha s = 10$	96

List of Tables

Table 3-1: Summary of definitions and notations used throughout this chapter	28
Table 3-2: 45-nm node technology parameters.....	31
Table 3-3: Detailed profile of α	33
Table 3-4: HSPICE simulation results of the two integrated bus structures (performance violations or sub-optimal results are underscored)	53
Table 3-5: Extracted parameters for the management of integrated buses.....	55
Table 3-6: HSPICE results for the two integrated buses subject to DVS (performance violations or sub-optimal results are underscored) M: Model.....	55
Table 4-1: Energy performance in platforms for three embedded applications.....	85
Table 4-2: Energy performance in enhanced platforms subject to DVS	85
Table 4-3: $TEP_{2-Error}$	85
Table 5-1: Power/energy profile of the TI ZigBee [®] WSN platform (Vdd=3V)	95
Table 5-2: Characterizing the TI ZigBee [®] WSN platform with four static activity states (energy unit is in uJ)	97
Table 5-3: Characterizing the TI ZigBee [®] WSN platform in cases with dynamic activities	various 98

List of Acronyms

ABB	Adaptive Body Biasing
ASIP	Application Specific Instruction-set Processor
CAS	Circuit And Systems
DPM	Dynamic Power Management
DSM	Deep Sub Micron
DFS	Dynamic Frequency Scaling
DVS	Dynamic Voltage Scaling
EMM (EM ²)	Energy Managed Microsystem
EP	Energy Performance
MCD	Multiple Clock Domains
MEP	Minimum Energy Point
MPSoC	Multi Processor System on Chip
NoC	Network on Chip
PLD	Portion of the Logic Delay
PMU	Power Management Unit
PWD	Portion of the Wire Delay
PVT	Process Voltage Temperature
SoC	System on Chip
SP	Speed Performance
ULP	Ultra Low Power
VLSI	Very Large Scale Integration
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Introduction

Energy consumption² is one of the key technological challenges that semiconductor industry has confronted in recent years. In the last decade, the transistor count per unit area has tremendously increased due to manufacturing advances. This has resulted into a very significant power density growth in modern electronic systems, especially in the high-performance computing segment. Figure 1-1 demonstrates the projections of the

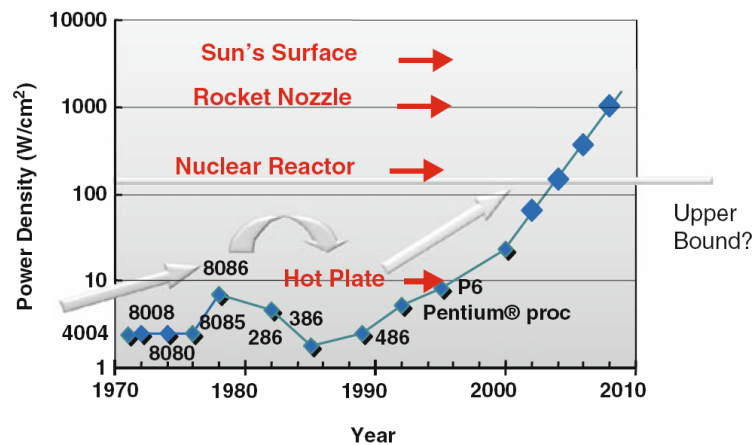


Figure 1-1: The power density trend of some well-known high-performance processors [2] [3].

² Energy consumption involves power consumption. Power is deduced from energy, and NOT vice-versa. Power, physically is defined by the rate at which energy is delivered, as defined by the following equation (P , E and t imply the components of power, energy and time, respectively) [1]:

$$P = \lim_{t \rightarrow 0} \Delta E / \Delta t = dE / dt$$

power density growth in some well-known high-performance processors -an important segment in semiconductor industry. When this figure was first introduced, projections were made to stress that if the observed trend continued at the same pace, future generations of these technologies would face inevitable difficulties. Integration technologies used without constraint can consume the kind of power predicted in Figure 1-1 but packages cannot dissipate the resulting heat and silicon based technology cannot operate reliably with junction temperatures above 125 °C. Needless to say that the observed power density in high performance integrated circuits have in fact saturated approximately at the levels observed in 2000.

In mobile computing (ranging from high-end laptops to ubiquitous sensory networks) - another emerging segment of the semiconductor industry- the energy resource (usually a battery) is limited. This limitation exists while (embedded) application complexities, driven by the demand of users, keep increasing with time. These requirements have resulted in many challenges in the design of energy-restricted mobile electronics. Figure

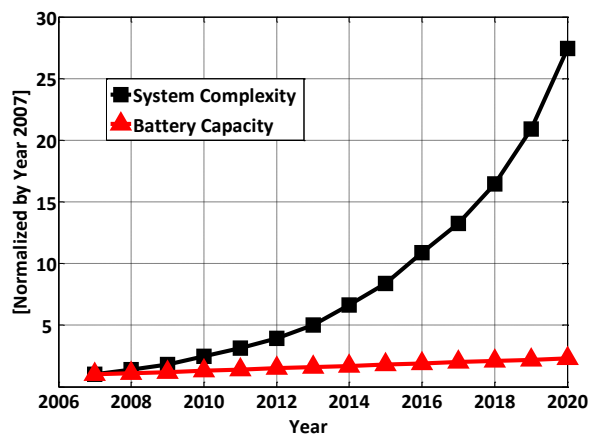


Figure 1-2: Mobile electronics complexity vs. battery capacity trends [4].

1-2 demonstrates the growth of computational complexity in mobile consumer electronics, as well as the system resource capacity. As can be seen from this figure, the system complexity, measured as the number of the processing engines (driven by system applications), is expected to grow 27 fold by 2020; whereas the energy resource budget is expected to grow only 2 fold, which correspond to a yearly growth of less than 10% [3] [4]. This shows the necessity of developing design methods for reducing energy consumption as the demand will obviously exceed the available energy budget.

Along with the feasibility concerns of future high-performance and mobile computing, environmental sustainability and environmental concerns of so-called “Green electronics”, are yet another important criterion, and therefore a new driver for *energy-aware design* of information processing systems [5].

The term “Green” initiated by the U.S. environmental protection agency [6] relates to products with high energy-efficiency. In this context, green computing was

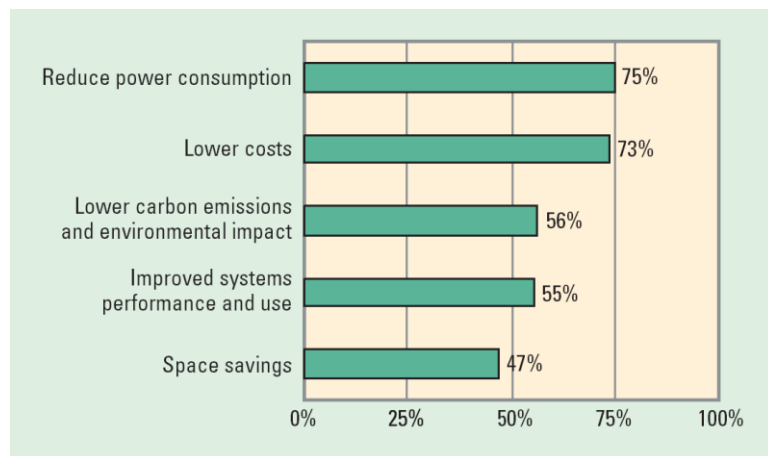


Figure 1-3: The benefit of designing green information processing systems [5].

correspondingly defined as “the study and practice of designing, manufacturing, using, and disposing of computers, servers, and associated subsystems-such as monitors, printers, storage devices, and networking and communications systems- efficiently and effectively with minimal or no impact on the environment” [5]. Green computing benefits the environment by improving the energy consumption, lowering greenhouse gas emissions, using less harmful materials, encouraging reuse and recycling, etc. [5]. A summary of such benefits is given in Figure 1-3.

Some major solutions to green information processing systems (sometimes referred to as green IT) are given in Figure 1-4. According to this figure, energy-efficient and power management are introduced as the first two in the list of important solutions for the design of green electronic systems provided in [5]. It is expected that the green criterion by itself may change the future design and usage models of information processing systems [7].

- Green IT spans a number of focus areas and activities, including
- design for environmental sustainability;
 - energy-efficient computing;
 - power management;
 - data center design, layout, and location;
 - server virtualization;
 - responsible disposal and recycling;
 - regulatory compliance;
 - green metrics, assessment tools, and methodology;
 - environment-related risk mitigation;
 - use of renewable energy sources; and
 - eco-labeling of IT products.

Figure 1-4: Some major solutions to green information processing systems [5].

1.2 Thesis Motivations

Based on the earlier discussions, energy management (which includes power management as well) is one of the unique solutions that simultaneously address the needs of high-performance, (ultra) low energy and greenness in computing systems. Based on the importance of such a design solution, the focus of this thesis is on the design exploration of Energy Managed Microsystems (EMS or EM^2), with the goal of improving their energy efficiency and/or operational lifetime. The proposed solutions are applicable to many classes of computing systems, including high-performance and mobile computing systems, and can contribute to make those systems “greener”. The proposed solutions are multilayer since they belong to, and some applicable to, multiple design abstraction layers.

1.3 Research problems, solutions and contributions

Numerous research problems relate to the design of EM^2 . In this thesis however, we tackled the ones that seemed more significant and fundamental to us. The research problems, the proposed solutions and contributions are formulated in the following.

1.3.1 Design exploration of EM^2 based on interconnection

Modeling, design and management (control) of interconnections in EM^2 has never been formulated in a comprehensive way. In Deep Sub Micron (DSM), as systems become interconnect-centric, accurate modeling and design of interconnections become very significant. For this purpose, interconnect-aware models that capture the performance of EM^2 (components, including interconnections) with good accuracy are proposed. The

reported results confirm the good accuracy of the proposed performance models. The proposed solutions were presented at GLSVLSI'09, in the form of the following paper:

Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "An Interconnect-Aware Delay Model for Dynamic Voltage Scaling in nm Technologies", In Proceedings of Great Lakes Symposium on VLSI (GLSVLSI), Boston, USA, May, 2009.

Repeater design and insertion, on the other hand, is a well-known method for performance improvement of interconnections. All the previously reported repeater insertion methods in the literature target systems operating at their nominal operating state. Accordingly, repeaters insertion methods for EM² interconnections are proposed. These solutions are submitted to GLSVLSI'11, in the form of the following paper:

Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "Repeater Insertion in Power-Managed VLSI", to appear in Great Lakes Symposium on VLSI (GLSVLSI), Lausanne, Switzerland, May, 2011.

An extended version of this work is submitted to the Journal on Emerging and Selected Topics in Circuits and Systems:

Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "Design Space Exploration of Interconnect Repeaters in Power-Managed VLSI", submitted to Journal on Emerging and Selected Topics in Circuits and Systems.

As a complement to the proposed modeling and design techniques, interconnect-aware management techniques for EM² (components, including interconnections) are proposed. The proposed methods demonstrate improved efficiency compared to available logic-

based management techniques. These solutions were presented at ISCAS'10, in the form of the following paper:

Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "An Interconnect-Aware Dynamic Voltage Scaling Scheme for DSM VLSI", In Proceedings of International Symposium on Circuit and Systems (ISCAS), Paris, France, May, 2010.

An extended version of the above contributions is submitted to Transaction on Circuit and Systems I:

Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "Modeling, Design and Management of Interconnects in Power-Managed VLSI", submitted to Transactions on Circuit and Systems.

1.3.2 Design exploration of EM^2 based on application

Modeling and design explorations of EM^2 are very valuable tasks which can accelerate the design process. Various modeling and design exploration methods of systems for speed performance optimization have been proposed in the literature. Amdahl's Law is one such popular high level abstract method. Reformulating Amdahl's law for the purpose of energy is valuable in order to help designers exploring EM^2 based on some specified target energy budget. In this thesis, models are proposed to estimate the energy consumption of EM^2 based on the application profile. These models work at high-level and therefore they can be computed rapidly with an acceptable accuracy. These solutions were presented at ICECS'09, in the form of the following paper:

Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "Estimation of Energy Performance in Computing Platforms", In Proceedings of International Conference on Electronics, Circuits and Systems (ICECS), Tunisia, Dec, 2009.

Energy estimation models are enhanced to consider EM^2 performing Dynamic Voltage Scaling (DVS). These high-level models demonstrate a fair accuracy. These solutions combined with results previously reported at ICECS'09 are submitted to IEEE Transactions on VLSI:

Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "Estimation of Energy Performance in Computing Platforms", submitted to Transactions on VLSI.

1.3.3 Design exploration of EM^2 based on activity

Modeling and design explorations of EM^2 will result into the acceleration of the design process. Based on our literature review, we noticed that no work has been performed for performance modeling and design exploration of EM^2 based on activity requirements. Accordingly, high-level models are proposed to estimate EM^2 energy, according to their activity profile. These solutions are submitted to GLSVLSI'11, in the form of the following paper:

Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "Activity Management in Battery-Powered Embedded Systems: A Case Study of ZigBee® WSN", Submitted VLSI-SoC, Oct, 2011.

In general, the proposed solutions tend to be orthogonal to each other. A priori, nothing prevents combining them. If these solutions were deployed simultaneously, the net

relative benefit would tend to be up to the order of the multiplication of the individual benefits.

1.4 Thesis organizations

The organization of this thesis is as follows.

In Chapter 2, the basics of power/energy consumption together with the review of the related literature are provided.

In Chapter 3, interconnect-centric techniques for modeling, design and management of EM^2 are given.

In Chapter 4, system-level models are proposed to estimate the energy consumption of EM^2 based on system application.

In Chapter 5, system-level models are proposed to estimate the energy consumption of EM^2 based on system activity.

In Chapter 6, the conclusions and the recommendations for future work of the thesis are provided.

Chapter 2

Energy Reduction Techniques in Various Design Abstraction Layers

2.1 Overview

In this chapter, we first review the basic concepts of power and energy, and later we review the literature on related energy reduction techniques that apply at various design abstraction layers. Performing an exhaustive literature review on energy reduction techniques is much broader than the scope of this chapter. Notably, this is the subject of recently published books, such as [3]. This chapter will thus focus on subjects and contribution sufficiently related to the content and contributions of this thesis.

2.2 Energy and power basics

Energy is defined as the effort needed to perform a task. Its unit is in Joule, and its components are Power (in Watts) and Time (in Seconds); wherein average power is defined as the energy consumed per time unit. Based on such a definition:

$$P = \frac{E}{T} \quad (1-1)$$

Accordingly, reduction of power tends to result directly into the reduction of energy. Thus, low power design techniques (assuming with no performance trade off) can be considered as a subclass of low energy design techniques.

Power consumption is mainly divided into two categories: *dynamic* and *static*. The difference between the two is that the former is proportional to system activity, whereas the latter is independent of activity. Until lately, the dynamic power in CMOS technology applied to digital circuits, the mainstream integration technology, surpassed the static power. Yet, with the emergence of nano-scale CMOS technologies, leakage has become a major power component; accordingly, both of these power components need to be considered with an equal importance.

In this thesis, we focus mainly on dynamic power/energy. Some techniques such as voltage scaling also improve the leakage power, but we mainly focus on the contributions which aim at reducing the dynamic component of system energy.

The energy driven from a supply-voltage source feeding a first-order RC network, subject to a step input, is [3]:

$$E = C V^2 \quad (1-2)$$

In which C is the value of network capacitance, and V is the value of the supply-voltage. An important note with respect to (1-2) is that the value of resistor does not affect the energy. In the design where an NMOS transistor replaces the resistance in the network, (1-2) can be represented as (assuming only the dynamic component of energy):

$$E_d = C V V_{NMOS} \quad (1-3)$$

In this equation V_{NMOS} is the output voltage of the charged NMOS transistor, which usually has the value of $V - V_{th}$, and V_{th} is the transistor threshold voltage.

Generalizing (1-3), assuming system (component) has a switching activity α_{sw} :

$$E_d = \alpha_{sw} C V V_{Swing} \quad (1-4)$$

In which V_{Swing} is the output voltage of a circuit/system that can be reached. Note that the maximum value of V_{Swing} can be the same as the supply-voltage V .

Defining P_d as the dynamic power consumption related to the energy consumption given in (1-4):

$$P_d = \frac{\alpha_{sw} C V V_{Swing}}{T} \quad (1-5)$$

$$P_d = \alpha_{sw} C V V_{Swing} f$$

In which $T = 1/f$, is the duration of the system functional activity. According to (1-4) and (1-5), *any solution that can reduce system activity α_{sw} , system complexity (resulting to lowering C), supply voltage V , output swing V_{Swing} or system frequency will reduce the dynamic power consumption.*

2.3 Minimum Energy Point (MEP)

As seen in the above formulations, the reduction of the supply-voltage has quadratic effect on the reduction of system dynamic power/energy (assuming full swing operation). Accordingly, static and/or dynamic low supply-voltage designs are highly desirable for energy-efficient system realizations. Due to this important fact, in this part, we briefly review the basics of deploying the minimum operating voltage. This fact needs to be considered for the design of Ultra Low Power (ULP) computing systems.

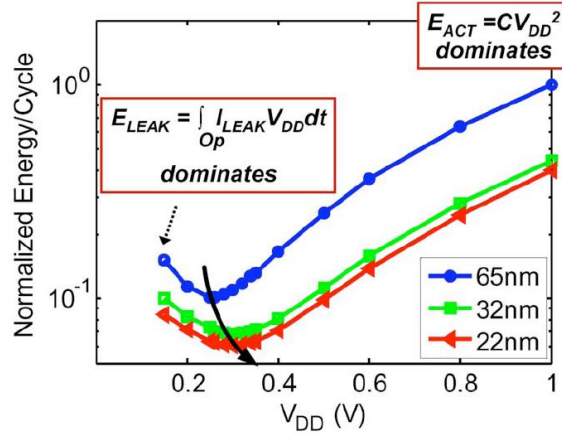


Figure 2-1: The minimum energy point trend in DSM [10].

Meindl et al. showed that the minimum allowed supply-voltage for a CMOS inverter to operate accurately (to be regenerative, i.e. gain ≥ 1 , and to have two distinct steady states “0” and “1”) at room temperature is 36mV [8]. This value has been introduced as the limit of supply-voltage scaling in CMOS technology.

Other researchers showed that for ULP Circuit and Systems (CAS) design, there is a design operating point that results into minimum computing energy. This design point may not necessarily have the minimum value of the supply-voltage allowed by the system [9]. This design point is referred to as Minimum Energy Point (MEP). This phenomenon is due to the fact that the leakage energy is linearly dependent with the CAS delay; accordingly the fraction of leakage energy may increase when the supply-voltage is in the sub-threshold regime³. In this case, even though the dynamic energy reduces quadratically along with the supply-voltage; in situations where dynamic and leakage

³ Sub-threshold region is defined as the region wherein the supply-voltage of the CAS is (close to, or) smaller than the threshold-voltage of a CMOS inverter [10]. Accordingly, the super-threshold region is defined as the region wherein the supply-voltage is larger than the threshold-voltage of a CMOS inverter.

energy components become comparable, the total energy increases with the supply-voltage scaling due to the increased circuit delay. This phenomenon can be seen in Figure 2-1.

The selection of MEP depends on the process technology, system architecture, application, workload, etc. [11] [12] [13]. Therefore a set of MEPs have been reported in the literature [9] [12] [13] [14]; all their values belong to the sub-threshold design region and are in the range of 200~400mV. It has been reported that major modifications need to be performed to standard cells, specifically for the case of integrated registers, to operate resiliently in this operating region [10] [11]. It has also been demonstrated that standard cells operate well with supply-voltages $\geq 400\text{mV}$. Consequently, for the purpose of this thesis, in order to benefit from the MEP concept without too much performance loss (also confirmed by Figure 2-1), and also from the fact that the standard cell libraries are widely available, we select the lower bound of the supply-voltage in our designs to 400mV.

2.4 Related energy reduction techniques

In this part, a survey of related power/energy reduction techniques is provided. Various techniques are grouped according to the level of abstraction to which they apply.

2.4.1 Circuit/System level techniques

Numerous related circuit/system level design solutions exist to achieve low-energy in VLSI computing systems. Employing VLSI systems with Dynamic Power Management (DPM) is one of the most effective methods that dynamically control the energy usage of

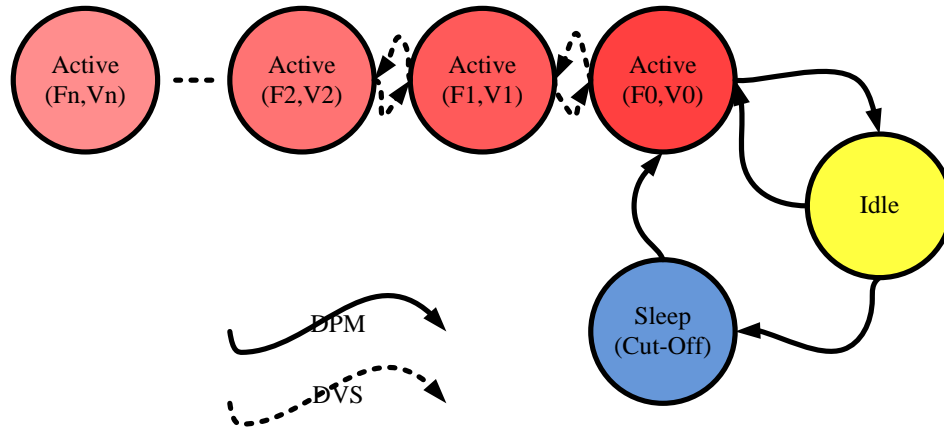


Figure 2-2: A model for a power-managed system performing simultaneous DPM and DVS (Inspired from [17]).

the system. In a power-managed system, a PMU dynamically controls the operating states of system components based on the application requirements [15]. This control is performed based on some policies. A large list of DPM policies is given in [16]. In DPM, usually by means of performing power gating and/or DVS and/or Dynamic Frequency Scaling (DFS) to system components, the total system energy is reduced. The simultaneous adoption of DVS and DPM was first introduced in [17]. In order to reduce system leakage power/energy, in platforms with power gating, the supply rail(s) dedicated to the processing elements are gated during sleep-mode [18] [19]. As the support of power gating may induce power-supply noise, the design time of a system that supports power gating increases significantly. Nevertheless, power gating has been widely deployed in industry, as it can significantly reduce the total system power/energy [20].

DVS and DFS on the other hand, help to further reduce system power/energy by changing system components operating voltage/frequency during their active-mode, to

meet “just-in-time” application requirements [21] [22] [23]. These techniques are widely employed in practice, as they are very effective methods, considering the fact that systems do not always need to operate at their peak performance. A power-managed system performing DPM and/or DVS may be modeled as in Figure 2-2. A number of modern commercial processors such as Intel XScale[®] [24], AMD Athlon[®] [25], IBM Power PC[®] [26] as well as Intel Pentium[®] [27] are equipped with DVS features.

Designing Application Specific Instruction-set Processor (ASIP), a successful class of configurable platforms, are yet another popular system/architectural option to improve processing energy efficiency. According to [28], computing platforms encompassing fixed and non-customizable components are poor choices when the design objectives are high performance, low energy consumption or reduced design time. Configurable technologies in the form of ASIPs are known to be the dominant trend to address these design objectives. Moreover, such customizable technologies provide on-chip and inter-module communications mechanisms that allow several fold improvement in their communications speed and energy [28]. In the case of mobile embedded systems and applications, where energy efficiency becomes the ruling factor, ASIPs achieve significant gains with respect to performance and energy. This is obtained by tailoring instruction-sets and micro-architectures according to the application requirements [29]. Numerous benefits and their related reason of adopting ASIP technologies are accordingly reviewed in [30] [31]. From this class, the Tensilica Xtensa processor technology is one of the most famous components that has been introduced and widely deployed in industry [32]. Some have implemented fully-customized ASIPs when more

energy-efficiency is required. A fully-customized ASIP for body-area WSN application has been developed in [33]. Some other customized ASIPs for more generic WSN applications often subject to tight energy constraints have been developed [34] [35] [36].

Parallelism and the deployment of Multi Processor System on Chip (MPSoC) and Network-on-Chip (NoC) -based MPSoC, is another effective method to alleviate high-complexity systems energy. Generally, partitioning a given set of tasks among several processing engines makes it possible for each engine to process its task with lower frequency -hence lower supply voltage requirement-, and therefore results directly to lowering the overall system energy. Numerous techniques and examples have been proposed where the overall processing energy is reduced by means of parallelism and multi processing [37] [38]. NoC-based MPSoCs, by means of supporting scalability in MPSoC design, help further to reduce the total system processing energy efficiency [39]. Accordingly, various design solutions have been addressed to design and improve communication-centric MPSoCs [39] [40].

MPSoC design in Deep Sub Micron (DSM) has become interconnect-centric, as centimeter-long interconnections are widely seen in these technologies [41] [42] [43]. Consequently, interconnects have become the main bottleneck against both the performance and energy efficiency improvements of these technologies. According to [41], more than 50% of the total chip power is consumed in interconnections, including the clocking. Accordingly, many efforts have been put toward the improvement of interconnection performance in DSM technologies. Many techniques have been proposed on interconnect repeater insertion in the literature, for interconnection performance

improvement [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54]. The optimal number and size of the repeaters to achieve minimum signal latency have been given in [44]. Some correlated contributions that investigate the design of interconnects with latency, operating frequency and area constraints, are found in [44] [45]. Some researchers have contributed to power-optimal interconnect synthesis considering latency and bandwidth constraints [46]. In [47], a buffer design has been proposed to reduce timing errors in interconnections subject to DVS. Some researchers have investigated the power-aware design of interconnect repeaters with latency constraints [48] [52]. Some have proposed boosted repeaters for long interconnect [49]. A technique for noise-aware repeater insertion has been proposed in [50]. Techniques for interconnection tree performance improvement have been proposed in [51]. The considerations of inductance effects into interconnect modeling has been extensively discussed in [53]. In contrast to uniform repeater insertion, a hybrid repeater insertion method has been introduced in [54]. All the previously reported methods however, consider systems running at their nominal operating state, whereas for the purpose of EM^2 realizations, interconnection design and repeater insertion methods should take the dynamic characteristics of EM^2 into account.

Many have contributed to improving the power/energy efficiency of DSM interconnect-centric power-managed MPSoCs. They have proposed techniques in which the communication and/or synchronization mechanisms are controlled independently from the computing engines. In [55], techniques have been proposed that dynamically reduce the on-chip communication router power. Techniques to apply DVS to clock networks in multiple clock domains have been proposed in [56]. Software-level DVS techniques have

been proposed that dynamically reduce the interconnection links energy in MPSoCs [57] [58]. In all the previously proposed techniques however, no explicit formulation has been proposed for interconnections realization in such dynamic systems. In order to fulfill such design objectives efficiently, the system elements involved with those interconnect-centric mechanisms, mainly as interconnects and associated repeaters need to be precisely designed, to meet the performance objectives, across various system operating states.

In the initial part of this thesis, we focus on techniques for modeling, design and management of interconnects in EM^2 to improve system energy efficiency. The proposed techniques take the dynamic characteristics of EM^2 into account. These contributions are given in Chapter 3.

2.4.2 System/Application level techniques

Numerous related system/application level contributions have also been proposed to achieve low energy in VLSI computing systems.

Contributions to system-level power/energy modeling of modern computing platforms have been reported. In [59] the authors propose power/energy estimation models for extensible instructions targeting mainly customizable processors, such as ASIPs. They consider power/energy overhead as well as latency based on a wide range of customization parameters. They employ system decomposition and regression analysis in order to achieve their goals. In [60] the authors propose a framework for analyzing, estimating and optimizing the processing engines power/energy at architectural level. They claim that their methods are 1000 fold faster than the low-level simulation analysis. In [61], the authors have presented a model for estimating power/energy based on a

constant parameter for power/energy consumption. They claim high accuracy while employing simple estimation technique based on software instruction execution time estimate and its average power. In [62] authors perform variation-aware system-level power analysis. They consider both manufacturing-induced (die-to-die and within-die) variations as well as dynamic variations during run-time because of the temperature effects.

Some have contributed to the system energy modeling based on software instruction profile [63] [64]. In [63], the authors propose a processor energy estimation technique, based on software instruction profile. They calculate the energy consumption of the software based on the current consumption and the execution time of each instruction. In [64], the authors propose energy macro modeling techniques that consider both the fixed and variable micro-architecture designs, such as in the case of conventional embedded processors, as well as the ASIPs.

In this thesis however, the proposed system-level energy models embed and abstract the architectural attributes to produce higher-level (i.e. system-level) energy estimation models.

Managing DVS at the Operating System (OS) or application level has been reported by several authors [65] [66] [67] [68] [69]. In [65] the authors propose a task scheduling algorithm -to perform DVS- to minimize energy. In [66] the authors describe the synthesis of SoCs based on the core processors while performing DVS. They treat DVS as a variable to be scheduled along with the computation tasks during scheduling. They also take into account the inherent limitation of the rate of deliverable DVS by the supply

voltage controllers and clock generators. In [67] the authors present an approach to find the minimal energy DVS schedule for executing real-time tasks according to some OS policies. In [68] the authors present a model for DVS processors as well as a static DVS scheduling formulation based on integer linear programming to minimize energy, under an execution time constraint. In [69] the authors propose an online-learning algorithm for system-level DPM and DVS. They formulate both DPM and DVS problems as a set of workload characterization and selection, and propose solutions based on their online-learning algorithm, accordingly.

Some fine-grained DVS techniques, considering deploying DVS to the individual blocks of an application -usually performed by compiler-, have been reported in [70] [71] [72] [73] [74] [75] [76] [77]. In [70] the authors present an intra-process DVS technique on the blocks of an application running on an embedded system platform. Their key idea is to make use of runtime profile information about the memory access in order to perform DVS at CPU. In [71] the authors propose a cooperative hardware-software control scheme for DVS to lower the power consumption of computing systems. In [72] the authors introduce an intra-task DVS technique for compiler control of application blocks. Their technique considers multiple intra-task performance deadlines and takes power budget into account. In [73] the authors propose a strategy for compiler management to perform DVS without significant overhead with respect to overall program execution time. In [74] the authors present a new concept of DVS for the purpose of multimedia applications. Such applications have a periodic property but with large variations with respect to their execution time. In [75] the authors present and review some techniques

for compiler control and task scheduling algorithms to perform DVS for the case of multiprocessors and MPSoC. In [76] the authors propose energy-aware task scheduling algorithm for applications on multiprocessor that exploits variations in the execution time of each individual task. In [77] the authors propose a platform that uses a pre-characterized model during run-time to predict the performance and energy consumption of a block of some application software.

In this thesis however, in Chapters 4, based on the application profile, we apply DVS to the segments of an application.

Application profiling is a well known method for speed and energy performance exploration in computing systems [78] [79] [80]. Speed and energy performance allow characterizing the net acceleration (or speed up) and energy savings of the computing platforms, obtained when some enhancements are applied to those platforms. Application profiling, for the purpose of speed performance estimation, was first introduced by Amdahl and ever since has been recognized as Amdahl's law [78]. Amdahl's law is a technique that estimates the expected speed-up, or the speed performance, of a parallelized implementation of an algorithm with respect to its sequential implementation. Due to the efficiency of this technique, some researchers used this technique for energy performance exploration of large scale systems. The authors of [80] present a framework for profiling server and cluster computers power consumption, for some parallel scientific applications, targeting high-performance distributed systems. Application profiling considers the fixed and tunable application segments; which can be adjusted to improve the speed and/or energy efficiencies.

To the best of our knowledge, no work has been reported on modeling of energy performance in computing platforms, including power-managed systems, considering software application segments.

In the domain of battery-powered embedded systems, the battery lifetime of a system is determined by its capacity as well as the energy drawn by the components attached to it. Since the advancement of battery capacity is much slower than improvements of microelectronic technologies [81], microelectronic design techniques that can contribute to increase the embedded systems battery lifetime are very valuable. In this domain, the authors of [81] present an overview of the emerging area of battery-aware design and exploration of embedded systems. They survey some promising technologies that are already developed for battery modeling and battery-aware system design. They also outline some emerging industry standards for smart battery systems.

The related solutions become even more critical, in the domain of Wireless Sensor Network (WSN) [82], wherein replacing batteries may not be feasible [83] [84] [85] [86] [87]. In [83], the authors present an empirical formulation of battery-powered WSNs based on the measurement of the current consumption of the processing nodes under different communication operations. In [84], the authors propose a modified Medium Access Control (MAC) protocol, called X-MAC, to make ZigBee protocol more power-aware and power efficient. In [85], the authors propose a method to estimate the lifetime of a WSN; given a network topology, a routing algorithm, an underlying MAC protocol and an initial energy budget. In [86], the authors propose a model to estimate the overall energy consumption of WSNs through their power dissipation. They employ

mathematical relations between the power dissipation during various operating modes. In [87], the authors introduce an ad-hoc routing protocol that require small memory footprint and takes advantage of extra memory resources to improve the quality of routing for critical WSN nodes.

Similarly, many battery-driven design techniques have been proposed to alleviate the energy consumption of battery-powered portable electronic systems, such as [88] [89] [90]. In [88], the authors survey various promising power-aware design techniques for both hard and soft real-time battery-powered embedded systems. In [89], the authors employ cross-layer optimization technique to propose energy efficient transport layer for wireless battery-powered embedded systems. They interplay between data aggregation, MAC and cooperative communications to achieve their goal. In [90], the authors compare the battery power efficiencies of various modulations schemes. They propose a closed-form analytical formulation for battery energy consumption accordingly.

We noticed from this review that no work has been performed for performance modeling and design exploration of embedded systems, EM^2 or WSNs, based on their activity. In this thesis, models are proposed that estimate the embedded system energy according to its “activity”. Good estimations of energy requirements are very useful when managing the system activity in order to extend the battery lifetime. These contributions are given in Chapter 5.

Chapter 3

Design Exploration of Energy Managed Microsystems Based on Interconnection

3.1 Chapter overview

This chapter proposes a complete flow which includes methods for modeling, design and management of on-chip interconnections, for power-managed VLSI. These methods guarantee that the designed and managed interconnects have *minimum* energy and area requirements, and that they meet all the performance objectives, here defined by operating frequencies and/or latencies, *in all the system operating states within system specification*. These methods take the impact of crosstalk into account. The contributions of this chapter are applicable to the design/management exploration of VLSI systems enabled with generic run-time power-management schemes, multiple voltage domains (voltage islands) and Multiple Clock Domains (MCD) systems, to name a few.

The contributions of this chapter are orthogonal to those from the other chapters; and if deployed simultaneously, the total net benefit may be the multiplication of the individual benefits.

3.2 Modeling interconnects in EM²

3.2.1 The interconnect performance models

For accurate performance modeling and analysis of on-chip interconnects in DSM power-managed VLSI, we consider their generic building block model. This model is based on the typical generic structure given in Figure 3-1. Based on this figure, the nominal propagation delay time ($T_{50\%}$) of a single interconnect stage in this structure can be expressed by (3-1) [91] [92]. Table 3-1 summarizes the definitions and the notations, used throughout this chapter.

$$T(l, w, h, k) = 0.69R_d(C_b + C_t/hk) + 0.38R_w(C_t/k^2 + 1.81C_g h/k) \quad (3-1)$$

Equation (3-1) shows that the interconnect delay is a function of the interconnect length (l) and its width (w), as well, as the size (h) and the number (k) of repeaters. In (3-1), the coefficient 0.38 converts RC time constants of a *distributed* RC delay model into a $T_{50\%}$ contribution to the wire delay, and the coefficient 0.69 represents the delay contribution for the rest of the *Lumped* components in the RC loops [91]. Equation (3-1) is a distributed line model and is based on the well-known Elmore delay model to compute

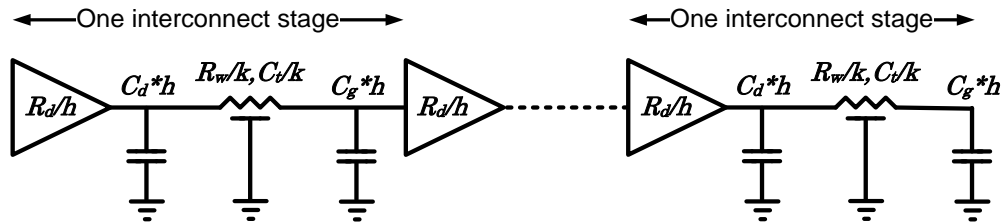


Figure 3-1: A generic decoupled interconnect model in Microsystems.

the device and interconnect delays [93]. Although the Elmore delay model may provide conservative delay estimation in DSM designs, for routing wire trees with many branches, it is still a good use as delay measurement for two-pin nets which includes the majority of all nets in real designs [94]; and thus for the focus of this work for high-level modeling, design and management purposes.

The effective interconnect capacitance (c_t) seen in the model is based on the consideration of the intrinsic interconnect capacitances (c_w) and neighboring coupling capacitances (c_c) multiplied by the effective switching factor Eta (η), as in (3-2). Equation (3-2) represents the decoupling technique that models the total effective capacitance c_t which is a widely accepted method of approximation for extracting the effective interconnects capacitances [95] [96] [97]. It has been shown that a typical value of 2 for η is practical when signal transition times is adequately fast [95] [98] [99]. The value of η however, in the presence of crosstalk in DSM designs, can reach the value of 4 [54] [97] [100].

$$c_t = c_w + \eta c_c \quad (3-2)$$

An important point with respect to the interconnect delay model given by (3-1), is the driver output resistance R_d , which is a supply-voltage dependent parameter [92]. Each operating state i is associated with an operating supply-voltage V_{ddi} , in this case the driver output resistance R_d , at the operating state i , is approximated by [92]:

$$R_{di} = \mu \left[\frac{V_{ddi}}{\underbrace{(V_{ddi} - V_t)^\alpha}_{N_i}} \right] \quad (3-3)$$

Table 3-1: Summary of definitions and notations used throughout this chapter

Notation	Meaning
l	Interconnect length
w	Interconnect width
h	Size of interconnect repeaters
k	Number of interconnect repeaters
R_d	Output resistance of a min-sized repeater
C_g	Input capacitance of a min-sized repeater
C_d	Output capacitance of a min-sized repeater
C_b	$C_b = C_g + C_d$
r_w	Interconnect resistance (unit length)
R_w	Interconnect resistance, $R_w = r_w * l$
c_w	Interconnect capacitance (unit length)
C_w	Interconnect capacitance, $C_w = c_w * l$
c_c	Interconnect coupling capacitance (unit length)
C_c	Interconnect coupling capacitance, $C_c = c_c * l$
η	Signal coupling switching factor
c_t	Effective interconnect capacitance (unit length), $c_t = c_w + \eta c_c$
C_t	Effective Interconnect capacitance, $C_t = c_t * l$
R_{di}	Output resistance of a min-sized repeater at system state i
V_{ddi}	The operating supply-voltage at system state i
V_t	Device threshold voltage
α	Device saturation index
μ	Device resistance coefficient
R_{d0}	Nominal output resistance of a min-sized repeater
s_i	System operating state i
S_i	Normalized scaling factor by which R_{di} scales with V_{ddi}
h_i	Size of repeaters, needed by system state i
k_i	Number of repeaters, needed by system state i
h_{il}	Size of repeaters, needed by state i , for latency objective
k_{il}	Number of repeaters, needed by state i , for latency objective
h_{if}	Size of repeaters, needed by state i , for frequency objective
k_{if}	Number of repeaters, needed by state i , for frequency objective
L_i	Interconnect latency at system state i
f_i	Interconnect operating frequency at system state i
α_s	Wire switching activity factor
ϑ	The product of h and k , $\vartheta = h * k$
N	Number of interconnects in parallel
A_R	Area overhead of a min-sized repeater
τ_d	Relative RC time constant due to repeaters
τ_w	Relative RC time constant due to interconnects

In (3-3), μ is a coefficient that incorporates the design and the technology parametric values. Equation (3-3) shows that the driver output resistance is supply-voltage dependant and varies with N_i . Accordingly, at the nominal operating conditions, $R_{d0} = \mu * N_0 (R_{d0}$

and N_0 represent the nominal values of R_{di} and N_i respectively). By substituting μ into (3-3), we conclude:

$$R_{di} = R_{d0} * S_i \quad (3-4)$$

Where

$$S_i = \frac{\frac{V_{ddi}}{(V_{ddi}-V_t)^\alpha}}{N_i} / \frac{\frac{V_{dd0}}{(V_{dd0}-V_t)^\alpha}}{N_0}$$

In (3-4), (N_i/N_0) is the *normalized scaling factor by which the output resistance R_{di} scales*, due to supply-voltage changes. We call this scaling factor S_i . Now, if we re-write (3-1) using the scaling factor S_i , we will have:

$$T_i(l, w, h, k) = 0.69[R_{d0}S_i](C_b + C_t/hk) + 0.38R_w(C_t/k^2 + 1.81C_g h/k) \quad (3-5)$$

Equation (3-5) is a line delay model that only reflects the *nominal design parameters*, assuming that other design elements are supply-voltage independent. Note that R_{d0} is the driver output resistance at the nominal supply-voltage for a given design, for which the key technology parameters can be obtained from vendors technology files for a given technology or can be obtained from ITRS [4], predictive models [101] [102], or by manual analysis.

In a power-managed Microsystem, each operating state i denoted by s_i , is associated with a different supply-voltage V_{ddi} (referred to as Dynamic Voltage Scaling (DVS)) and/or with a different operating frequency f_i (referred to as Dynamic Frequency Scaling (DFS)). Now assuming that in a power-managed Microsystem subject to operate at

multiple states, at each state, various design attributes for repeaters (i.e. h and k), may be needed to satisfy some performance objectives, in this case, the given interconnect performance model (3-5), can be modified to:

$$T_i(l, w, h_i, k_i) = 0.69 \underbrace{R_{d0}(C_b + C_t/h_i k_i)}_{\tau_d} S_i + 0.38 \underbrace{R_w(C_t/k_i^2 + 1.81C_g h_i/k_i)}_{\tau_w} \quad (3-6)$$

The difference between the performance models (3-5) and (3-6) is that in the latter, distinct repeater design attributes (h_i, k_i) are coupled with operating state s_i .

In (3-6), τ_d and τ_w respectively represent the relative RC time constants due to repeaters and interconnect. Normalizing (dividing) the performance model (3-6) by its value at the nominal operating state ($T_0 = T_{i|i=0}$), regardless of the repeaters design attributes (i.e. (h_i, k_i)), yields:

$$T_{ni}(l, w, h_i, k_i) = \underbrace{\left(\frac{1}{1+0.55\frac{\tau_w}{\tau_d}} \right)}_{\tau_{d-r}} S_i + \underbrace{\left(\frac{1}{1+1.81\frac{\tau_d}{\tau_w}} \right)}_{\tau_{w-r}} \quad (3-7)$$

In (3-7) the interconnect/wire delay portion (τ_{d-r}) and the repeater (driver) delay portion (τ_{w-r}) are obtained by means of dividing the delay components τ_d and τ_w , by the total delay time at the nominal operating state (assuming a same design).

The obtained performance models (3-6) and (3-7) will be utilized later on, for the design and management of interconnects in power-managed Microsystems.

3.2.2 The impact of technology parameters on the performance models

A 45-nm technology node based on the Berkeley Predictive Technology Model (BPTM) is used in this chapter for performance modeling [101] [102]. The associated parameters for the adopted technology are given in Table 3-2 [94] [101] [102] [103].

The interconnect repeaters considered in this chapter, are to be implemented as CMOS inverters. The repeaters are designed to be symmetrical with respect to their rising and falling signal transition times. The PMOS to NMOS transistor ratio, for this design technology, is considered to be 2.5.

The supply operating range is considered to be 0.4-1.0V which implies perfect super-threshold operating range with no need for standard device library optimizations or modifications [10] [11]. It has been reported that 0.4V supply voltage is a solution that delivers (close to) energy optimal design, especially adopted for the synthesis of low duty-cycle applications where mainly require ultra low voltage operations where ultra low voltage operation is required [9] [10] [11] [12].

Another important parameter to be discussed is α ; the velocity saturation index. The value of α in typical full-swing super-threshold designs is smaller than 2 [91]. In the case

Table 3-2: 45-nm node technology parameters

V_{dd0} (V)	1.0
V_t (V)	0.292
r_w (Ω/mm)	129
c_w (fF/mm)	44
c_c (fF/mm)	10
R_{d0} ($\text{K}\Omega$)	15.7
c_g (fF)	0.45
c_d (fF)	0.41
A_R (μm^2)	0.034

of near-threshold (approaching sub-threshold) designs, the value of α is better approximated by a value close to 2 [92]. In general, there is no definite closed-form value or exact formulation for α , and its value varies according to process technology, design parameters, as well as *the supply-voltage* [92] [104]. For this work, we performed some simulation-based analyses with the selected technology, in order to extract the correct profile of α .

The timing behavior for a cascade of minimally-sized symmetrical inverters for the given operating range 0.4V to 1V has been depicted in Figure 3-2 subject to a step input stimuli. We used the performance model (3-7), to extract the approximate value for α from HSPICE simulation results. When the circuit is at the nominal supply voltage, i.e. 1V, the value of α that provides the best approximation is $\alpha \approx 1.2$. It is of interest however, that *the value of α , providing the best performance fit increases as the supply voltage decreases*. According to Figure 3-2, $\alpha \approx 2$ is close to the best value to predict

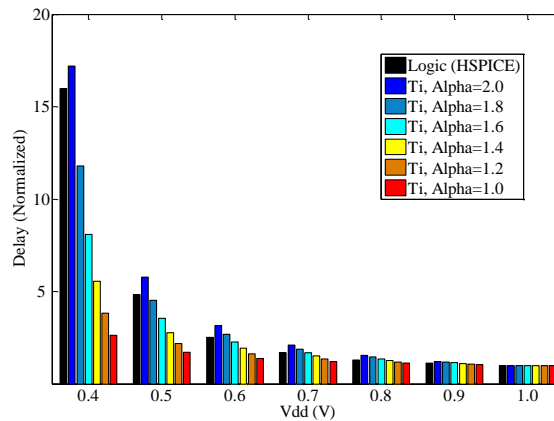


Figure 3-2: Extracting the approximate value for velocity saturation index α that best matches HSPICE transient simulation results. The delays are normalized by their respective nominal values at $V_{dd}=1$.

delay scaling over a range of supply voltages when the supply is scaled from 1V to 0.4V. Note that smaller values of α can lead to predicted delay scaling much lower than simulated delays when V_{dd} is reduced. The values of α that provide the best delay approximations with (3-7) as compared to the full circuit simulation when V_{dd} is varied between 0.4-1V are given in Table 3-3. This information is useful when modeling, designing and managing power-managed VLSI Microsystems.

An important fact with respect to Table 3-3 is that, the (α, V_{dd}) pairs in this table are coupled (since α is a function of V_{dd}); that means for each supply voltage operating region there is an associate α ; if in the analysis, the obtained pair does not comply with Table 3-3, the analysis should be repeated based on the fundamental variable, i.e. V_{dd} . A third-degree curve-fitted polynomial function that characterizes the relation of V_{dd} and α , according to Table 3-3, can be represented by:

$$F_{\alpha}(V_{dd}) = -5.55V_{dd}^3 + 12.38V_{dd}^2 - 9.99V_{dd} + 4.37 \quad (3-8)$$

Table 3-3 characterizes the relation between V_{dd} and α that is obtained from HSPICE, as well as, the quantified value of the proposed curve-fitted functional model F_{α} . Either of

Table 3-3: Detailed profile of α

V_{dd} (V)	α (HSPICE)	F_{α} (Model)
0.4	2.0	2.00
0.5	1.8	1.78
0.6	1.6	1.63
0.7	1.6	1.54
0.8	1.4	1.46
0.9	1.4	1.36
1.0	1.2	1.20

these relations, i.e. (α, V_{dd}) or $(F_{\alpha}(V_{dd}), V_{dd})$, can be utilized in the processes of modeling, design and management of interconnects in power-managed systems.

According to Figure 3-2, some residual error exists for all values of α ; assume to hold over the same operating range, for performance estimation. Figure 3-3 depicts the average error observed, when different values for α are used for estimating performances *over the full operating range*. According to Figure 3-3, $\alpha \approx 1.8$ produces the minimum average error (less than 4% smaller than the error observed when $\alpha=2$) over the full operating range 0.4-1V. However, $\alpha \approx 2$, provides the best approximation of the maximum possible scaling (obtained when supply voltage is at its minimum value, here 0.4V), which is a very important design characteristic, especially for the synthesis of *low duty-cycle applications*. Figure 3-3 can help for modeling, design and management of power-managed systems, when a single value of saturation index, need to be used.

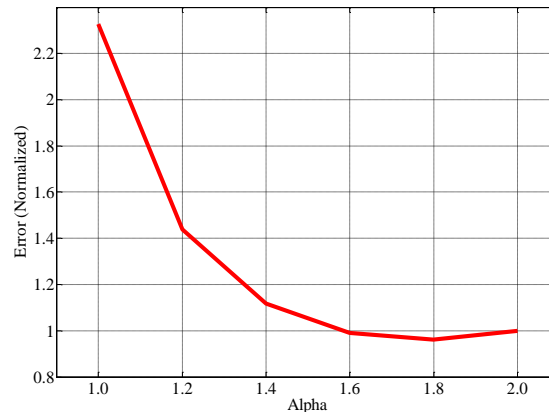


Figure 3-3: Average error when the performance is estimated with different values of α , assumed to hold over the full operating range 0.4-1V, normalized with respect to $\alpha=2$.

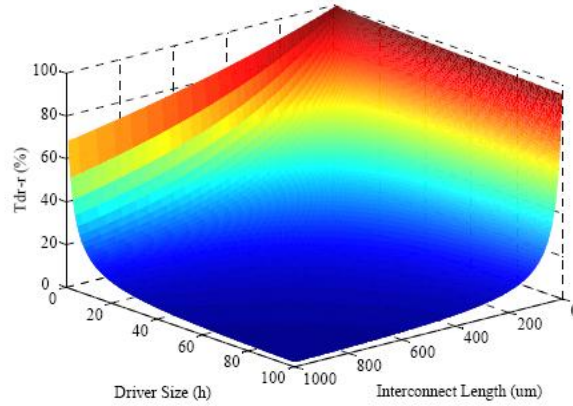


Figure 3-4: The driver delay portion, obtained based on the driver size (h) and the global interconnect length in 45-nm ($k=1, \alpha=2$).

3.2.3 The implications of the performance models

Figure 3-4 depicts τ_{d-r} in interconnect-centric designs based on the same 45-nm technology node using (3-7). According to Figure 3-4, *as interconnects become longer and the driver (repeater) size increases, τ_{d-r} tends to decrease*. Now, Figure 3-5 reports T_{ni} as a function of V_{dd} and τ_{d-r} . In the domain depicted in the figure, T_{ni} passes 17 for logic (e.g. an ALU), whereas for a typical interconnect-centric sub-/system (e.g. a bus) with a $\tau_{w-r}=50\%$ ($\tau_{d-r}=50\%$), T_{ni} drops below 10. It is of interest that based on Equation (3-7), smaller τ_{d-r} implies less delay scaling with voltage. Thus, *in advanced nanometer technologies, as wires become longer and the frequency of operation increases, the performance of Microsystems does not scale down as much with voltage scaling*.

3.3 Designing interconnects in EM²

In this section, utilizing the performance models given earlier, we propose methods to explore the design space for interconnect in power-managed Microsystems. These

methods guarantee that the designed interconnects will meet the performance objectives in all the system operating states, given by the system specification.

3.3.1 Designing interconnects with latency objectives

Let us consider a power-managed Microsystem, projected to run at a given set of operating states with certain latency objectives, which can be posed by system cycle-time constraints, and is denoted by $\{(L_i, V_{ddi}) \mid i \in \text{operating states } 0 \dots n\}$. For this, the interconnect repeaters should be designed in a way to meet the target interconnect latency objectives at the projected system processing states.

The line latency is defined as the delay between the two ends of the interconnect and therefore is k times the delay of one line segment. In this case, this latency in power-managed systems at the operating state i , is modeled by:

$$L_i \geq k_i * T_i \tag{3-9}$$

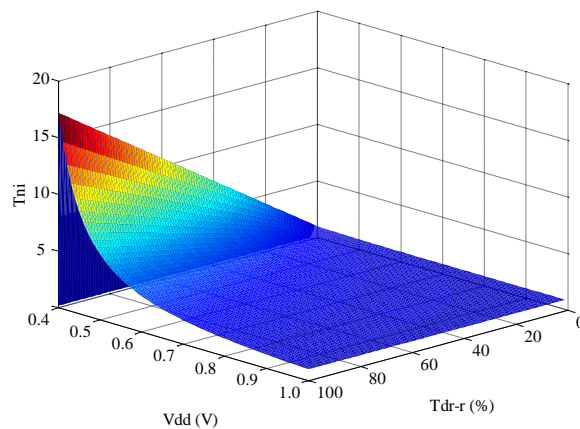


Figure 3-5: Sub-/systems delay subject to voltage scaling, based on the driver delay portion (τ_{d-r}) and the supply-voltage, ($k = 1, \alpha = 2$).

In (3-9), T_i replaces (3-6). Equation (3-9), gives $\{(h_{il}, k_{il}) | i \in \text{operating states } 0 \dots n\}$ that characterizes the repeater design space for which target interconnect latencies are met at the operating state i . Note that, according to this equation, the obtained solutions are *upper-bounded*, since $k_i \leq L_i / T_i$. The interconnect latency objectives must be met in all the design operating states. Thus, the latency-aware interconnect design sub-space in a power-managed system is obtained through:

$$\{(h_{PM-L}, k_{PM-L})\} = \{(h_{0L}, k_{0L})\} \cap \dots \cap \{(h_{nL}, k_{nL})\} \quad (3-10)$$

3.3.2 Designing interconnects with frequency objectives

Let us consider a power-managed Microsystem, projected to operate at a given set of operating states with certain operating frequency objectives, which can be posed based on system workload model, and is denoted by $\{(f_i, V_{ddi}) | i \in \text{operating states } 0 \dots n\}$. For this, the interconnect repeaters should be designed to meet the target operating frequencies at the projected system operating states. Operating frequency relates to the frequency of one single stage of interconnect. We target interconnect operating frequency instead of interconnect bandwidth, since our analyses are based on $T_{50\%}$, whereas for bandwidth, the analyses are based on signal full transition time [46]. Note that as defined, the latency and bandwidth constraints are decoupled, as multiple bus cycle time (inverse of bus frequency) may fit in one bus latency time. This model covers the case where a bus is wave pipelined with multiple bits transmitted in a single latency time.

To solve this design problem, (3-6) is adopted to extract the required design values (h_{if}, k_{if}) that enable interconnects to meet target operating frequencies $\geq f_i$ at the

operating state i . Solving (3-6) for k_i as a function of h_i , assuming the operating state i , results into:

$$k_i = K(h_i) = \frac{\beta + \sqrt{\beta^2 - \gamma R_w C_t}}{-2\gamma} \quad (3-11)$$

Where:

$$\beta = 0.69[(R_{d0} * S_i)C_t/h_i + h_i R_w C_g]$$

$$\gamma = 0.69(R_{d0} * S_i)C_b - T_i$$

In (3-11) $T_i=1/f_i$. The solution gives $\{(h_{if}, k_{if})|i \in \text{operating states } 0 \dots n\}$ that defines the design space. Note that the *lower-bound* of the design space is obtained according to f_i and the design space for operating frequencies $> f_i$ is also valid. In order to ensure that interconnect operating frequency objectives are met in all the design operating states, the solution should belong to the obtained solutions of all the operating states. In this case, the interconnect design sub-space to meet the operating frequency objective, in the power-managed systems becomes:

$$\{(h_{PM-f}, k_{PM-f})\} = \{(h_{of}, k_{of})\} \cap \dots \cap \{(h_{nf}, k_{nf})\} \quad (3-12)$$

Depending on the interconnect design objectives, *the valid portion of the design space* $\{(h_{PM}, k_{PM})\}$ for interconnect repeaters in power-managed systems is defined by (3-10) when the design is only constrained by latency objectives (case I). It is defined by (3-12) when the only constraints are operating frequency objectives (case II). Finally, the

intersection of the two, defines the space of valid solutions when both sets of objectives must be met simultaneously (case III). In other words:

$$\{(h_{PM}, k_{PM})\} = \begin{cases} \{(h_{PM-L}, k_{PM-L})\} & \text{(I)} \\ \{(h_{PM-f}, k_{PM-f})\} & \text{(II)} \\ \{(h_{PM-L}, k_{PM-L})\} \cap \{(h_{PM-f}, k_{PM-f})\} & \text{(III)} \end{cases} \quad (3-13)$$

3.3.3 Designing interconnects with energy objectives

The line average energy model in a power-managed VLSI system, running at the operating state i , in a unified form, is given by [105] [106] [107]:

$$E_i = \alpha_s [C_t + \vartheta_{PM} C_b] V_{ddi}^2 \quad (3-14)$$

In this equation, α_s denotes the line (wire) switching activity factor and $\vartheta_{PM} = h_{PM} k_{PM}$. The only design variable in this model is ϑ_{PM} . Also, as can be seen from (3-14), the line energy requirement increases monotonically with ϑ_{PM} , since $\partial E_i / \partial \vartheta_{PM}$ is a positive value. *Therefore the minimum energy is obtained for the minimum value of ϑ_{PM} in the valid portion of the design space $\{(h_{PM}, k_{PM})\}$.*

3.3.4 Designing interconnects with area objectives

The spanning area overhead due to repeater insertion in interconnects have been reported as [45] [103]:

$$A = N \vartheta_{PM} A_R \quad (3-15)$$

In which N is the number of interconnects in parallel, A_R is the area overhead of a minimum sized repeater defined in a given technology, and $\vartheta_{PM} = h_{PM} k_{PM}$. The only

design variable in this model is ϑ_{PM} . Also, as can be seen from (3-15), the area also increases monotonically with ϑ_{PM} , since $\partial A/\partial \vartheta_{PM}$ is a positive value. *Therefore the minimum area overhead in interconnects, is similarly obtained for the minimum value of ϑ_{PM} in the valid portion of the design space $\{(h_{PM}, k_{PM})\}$.*

Due to the above observations, we consider ϑ_{PM} as a *figure of merit* for energy-optimal and area-optimal design solution extraction from the interconnect design space $\{(h_{PM}, k_{PM})\}$ for power-managed VLSI.

3.4 Managing interconnects in EM²

In the previous section, design methods were introduced that guarantee the design of interconnects that meet the system performance objectives of all the system operating states. The goal of this section is to propose methods to manage/control the designed interconnect in power-managed Microsystems for efficient operation. DVS conventionally is performed using performance models derived for CMOS logic [9] [21] [108] [109] [110]. Consequently, the associated DVS schemes should be customized to maintain accuracy. In this part, for the purpose of interconnect management, a DVS scheme is proposed that considers the effect of interconnect parasitic into account. To fulfill this, some design metrics and formulations are initially introduced.

3.4.1 The DVS design metrics

According to (3-7), each distinct segment in a power-managed system with scaling rate S_i (at the operating state i), has an effective scaling rate proportional to $\tau_{d-r} * S_i$. Since S_i is the global scaling rate which applies uniformly to the system component, then τ_{d-r}

becomes *distinct for each interconnect performance* subject to dynamic state transition. We call the factor τ_{d-r} , *PLD*, which designates the *Portion of the Logic Delay (PLD)* in that interconnect segment. Accordingly, *PLD* is considered as a *design metric*, which determines the performance of an interconnect-centric component in a power-managed system subject to dynamic state transitions.

According to (3-7), each segment has also a unique complementary τ_{w-r} term. Similarly, we call τ_{w-r} , *PWD*, the *Portion of the Wire Delay (PWD)* in that interconnect segment. This concept can be formulated by (3-16). Accordingly, *PWD* could be also considered a *design metric*. Note that either metrics may be employed, as they are complementary.

$$PLD + PWD = 1 \quad (3-16)$$

3.4.2 A compact DVS model using the design metrics

Based on the above formulations, the delay model given by (3-7) using design metrics *PLD* and *PWD*, as well as (3-16), may be reformulated:

$$T_{ni} = PLD * S_i + [1 - PLD]$$

$$T_{ni} = PLD * [S_i - 1] + 1 \quad (3-17)$$

3.4.3 Scaling limit for error-free system operation

In VLSI, wire delays do not scale with the supply voltage (at least in its first approximation [92]). This implies that the system scaling (T_{ni}) is limited by interconnects. Therefore, in a Microsystem subject to DVS, *the segment that has the maximum interconnect delay portion* (i.e. maximum *PWD* and hence the minimum *PLD*),

becomes the bottleneck against error-free scaling (assuming a minimum value is defined for V_{ddi} which results into a maximum value for S_i). Utilizing (3-17), this limit can be formulated as:

$$T_{n-limit} = PLD_{min} * [S_{max} - 1] + 1 \quad (3-18)$$

Where PLD_{min} is the PLD of the segment in the system which has the minimum logic delay portion. In this equation, $T_{n-limit}$ indicates the scaling limit that can be reached by that DVS Microsystem (or its component) *to avoid incorrect operation*. Note that the scaling rate S_{max} is obtained with the supply voltage set at its minimum acceptable value. The minimum acceptable value for the supply voltage can be determined by the design and/or the process technology. For system components that mainly include logic like an ALU, $PLD_{min} \cong 1$ and therefore $T_{n-limit} \cong S_{max}$. *This is the same limit posed by the given process technology*. In interconnect-centric components such as in CDNs, NoCs or buses, $PLD_{min} \ll 1$ and therefore $T_{n-limit} \ll S_{max}$; therefore, *in interconnect-centric components, the system scaling limit can reach values significantly lower than the limit dictated by the process technology*.

3.4.4 Selecting supply voltages

Recalling from Section 3.4.2, (3-17) is a reformulated delay model by which a target delay (or a target frequency) may be obtained. Equation (3-17) can be considered as a function of independent variables. Such function, in its most concise form, is defined by:

$$T_{ni} = F_{Ti}(PLD, V_{ddi})$$

In the definition of function F_{Ti} , it is assumed that PLD and V_{ddi} are considered to be the only variables on which the function depends. Now, if we compute the inverse of F_{Ti} w.r.t. V_{ddi} , we will have:

$$V_{ddi} = F_{Ti|V_{ddi}}^{-1} = F_{Vi}(PLD, T_{ni}) \quad (3-19)$$

The function F_{Vi} defines a relation by which the supply voltage, based on a target delay and a given PLD , is obtained. *The obtained supply voltage, based on (3-19), is the solution for precise/efficient interconnect functionality at the operating state i . This supply voltage may be different than the one initially introduced by the system specifications which mainly consider logic.* An important fact about F_{Vi} is that its formation is heavily influenced by the value of α . When α equals to 1 or 2, the resulting quadratic expression has a tractable analytic solution expressed by (3-20) and (3-21).

3.4.5 The interpolation method

Formulating F_{Vi} for $1 < \alpha < 2$ is not a trivial task. In such cases, we may leverage some form of interpolation for quantifying F_{Vi} (i.e. quantifying V_{ddi} when $1.0 < \alpha < 2.0$). For this, we initially explore the two bounds, i.e. the behaviour of (3-20) and (3-21), for a large spectrum of designs in 45nm technology.

$$F_{Vi|\alpha=1} = \frac{N_0 V_t (T_{ni} + PLD - 1)}{N_0 (T_{ni} + PLD - 1) - PLD} \quad (3-20)$$

$$F_{Vi|\alpha=2} = \frac{PLD + 2 N_0 T_{ni} V_t + 2 PLD N_0 V_t - 2 N_0 V_t + \sqrt{PLD^2 + 4 PLD N_0 T_{ni} V_t + 4 PLD^2 N_0 V_t - 4 PLD N_0 V_t}}{2 N_0 (T_{ni} + PLD - 1)} \quad (3-21)$$

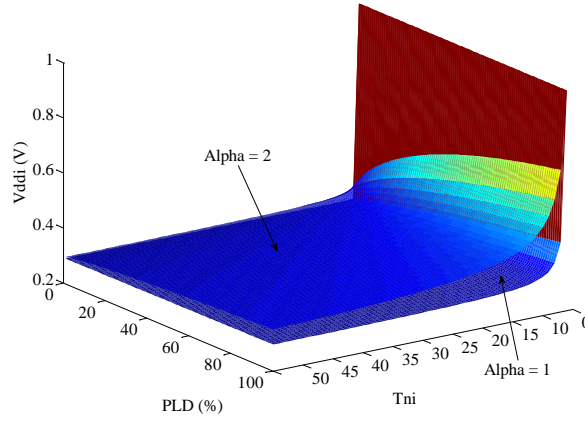


Figure 3-6: Quantifying F_{Vi} for a large spectrum of designs for $\alpha = 1$ and $\alpha = 2$ in 45nm technology.

Figure 3-6 depicts this exploration. According to this figure, the two configurations for α equal to 1 and 2 create bounds for trajectories with similar behavior, for a wide range of given design points (PLD, T_{ni}). Because of this behavior, averaging as a means for interpolation is attractive due to its simplicity and its efficiency. The procedure for formulating F_{Vi} for $1 < \alpha < 2$ is as follows. Knowing that F_{Vi} , based on the technology defined parameters and for α equal to 1 and 2 is bounded (referring to Figure 3-6), we first define α_1 and α_2 to be in the range from 1 to 2 (as relevant to the context). Thus, they are also bounded and therefore their average, i.e. $(\alpha_1 + \alpha_2)/2$, is in the same range and is bounded. Accordingly, F_{Vi} for α equal to $(\alpha_1 + \alpha_2)/2$ is bounded. In this case, we interpolate F_{Vi} by:

$$F_{Vi}|_{\alpha=\left(\frac{\alpha_1+\alpha_2}{2}\right)} \approx \frac{F_{Vi}|_{\alpha=\alpha_1} + F_{Vi}|_{\alpha=\alpha_2}}{2} \quad (3-22)$$

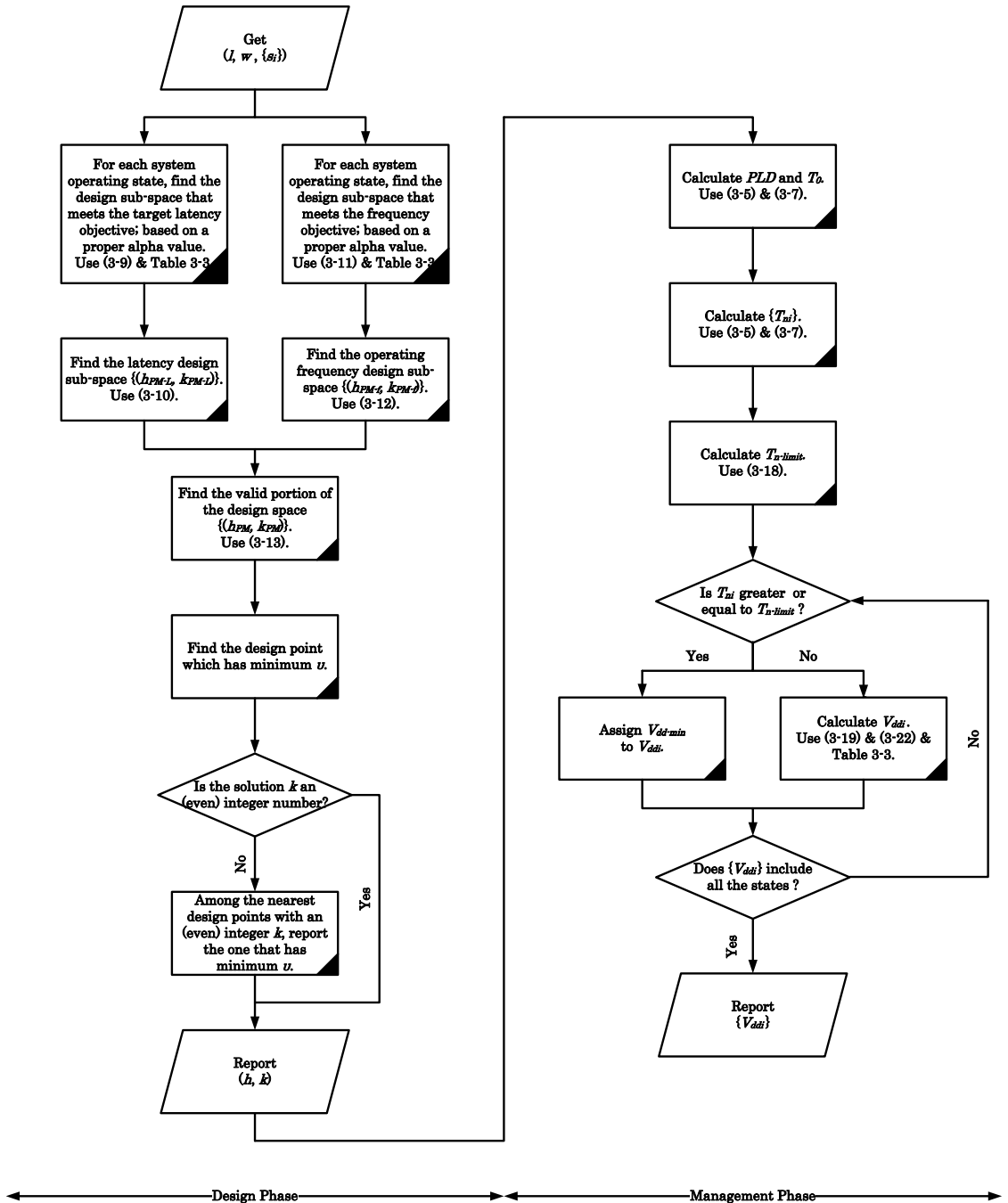


Figure 3-7: The interconnect design and management flow.

We may use (3-22) to quantify F_{Vi} for diverse quantities of α when $1 < \alpha < 2$. We may commence the iterative process of the averaging method with α equal to 1 and 2, utilizing (3-20) and (3-21), and iterate to converge to the desired α .

3.5 The flow

The proposed modeling, design and management methods are encompassed in a complete flow with two main phases: phase #1 including the design and phase #2 including the management of interconnects, as depicted by Figure 3-7.

In the design phase, based on the modeling and design methods proposed in Sections 3.3.1 and 3.3.2, the valid portion of the design space for interconnects is explored. According to Figure 3-7, initially the flow accepts the interconnect length l and width w of the wire, as well as the set of the system operating states $\{s_i\}$ with their design objectives $\{s_i = (L_i, f_i, V_{ddi}) \mid i \in \text{operating states } 0..n\}$. Later, utilizing the proposed models, the two design sub-spaces for interconnects which meet the latency and operating frequency objectives, $\{(h_{PM-L}, k_{PM-L})\}$ (using (3-9) and Table 3-3), and, $\{(h_{PM-f}, k_{PM-f})\}$ (using (3-11) and Table 3-3), are obtained. Later, the valid portion of the system interconnects design space $\{(h_{PM}, k_{PM})\}$, based on either or both of these design sub-spaces (using (3-13)) is deduced. In the valid portion of the interconnect design space $\{(h_{PM-f}, k_{PM-f})\}$, the optimal design point, with respect to both energy and area efficiencies, is the one which has minimum $\vartheta (h * k)$ value. In practice, the number of interconnect repeaters (k) should be an integer number; and when no bit inversion is required, k should be an even integer number. Since, the analytical formulations may result into solutions that do not meet this objective; eventually, the obtained solution may need to be updated. For this purpose, the design point closest to the optimal design point in the design space $\{(h_{PM}, k_{PM})\}$ that has an (even) integer k , with the smallest ϑ , is

reported as the design space exploration solution. At this point, the flow delivers the solution (h, k) of the given interconnect.

In the management phase, based on the solution (h, k) obtained from the design phase, initially the nominal interconnect delay (using (3-5)) and its *PLD* (using (3-7)) are both calculated. The required scaling at the operating states, based on (3-5) and (3-7) are obtained afterwards. Later, the limit of scaling, which could be reached by the given interconnect is calculated, using (3-18). This limit is important for the design cases where timing violations should be avoided, especially in the cases of critical paths, race-free sub-/systems, etc. Afterwards, this limit is compared with the given system scaling requirements. For each operating state i , if the required performance is not greater than the scaling limit permitted by interconnect, the supply voltage, using (3-19) and (3-22) when needed, is obtained; otherwise, the minimum defined supply voltage is assigned. At this point, the flow delivers the solution $\{V_{ddi}\}$, which implies the set of supply voltages required by that interconnect in all the system operating states.

3.6 Case study: design and management of integrated multi-cycle buses in a power-managed system

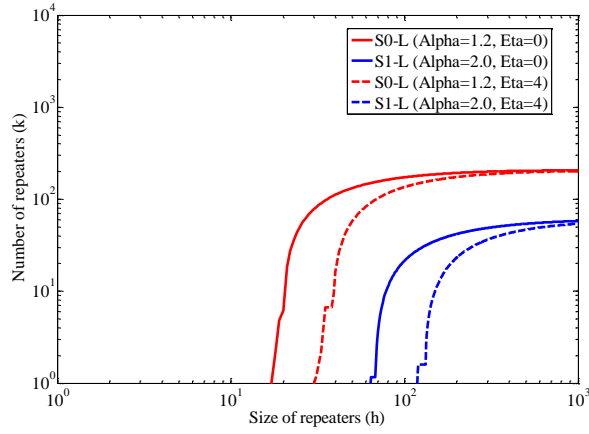
In this section, in order to validate the proposed design and management schemes, a design is chosen as a case study. We employ the proposed methods for the design and management of integrated buses, in a power-managed platform. HSPICE simulations are performed to confirm the validity of the proposed methods.

Let's consider a power-managed system, running at two distinct active operating states, for which the operating frequency and latency objectives, as well as the logic operating states supply voltages are specified as: $\{s_0 = (2\text{GHz}, 2\text{ns}, 1\text{V}), s_1 = (500\text{MHz}, 10\text{ns}, 400\text{mV})\}$. The target communication mechanisms are two integrated 8-bit buses whose length and width are given as Bus #1 (10mm, 8X) and Bus #2 (10mm, 4X), which aim to meet the specified performance objectives, given earlier. Bus #1 is twice wide as Bus #2. The bus design model follows the methods adopted in [111] [112] [113] [114].

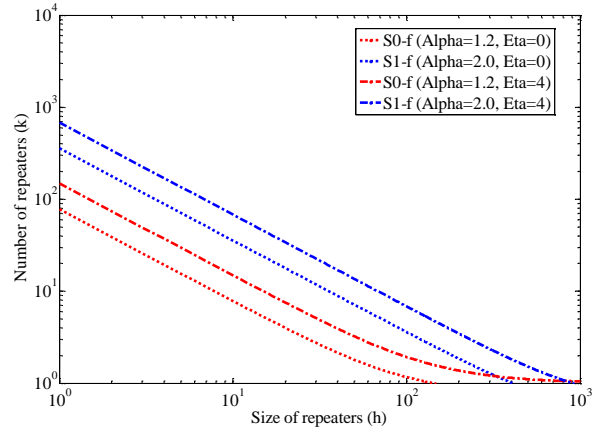
A 45nm technology node, with the parameters given in Table 3-2, is adopted. Table 3-3 has been used to employ the suitable values for α in each operating state. It is also assumed that only even number of repeaters (k) can be used (no bus inversion).

We considered wide global wiring structures, normally used in the design of synchronization or communication mechanisms in VLSI systems. We first perform the design space exploration for a bus with $8 \times w_{\min}$ line widths. w_{\min} is the minimum wire width defined by the process technology. In the second design, we consider the bus line widths reduced to half.

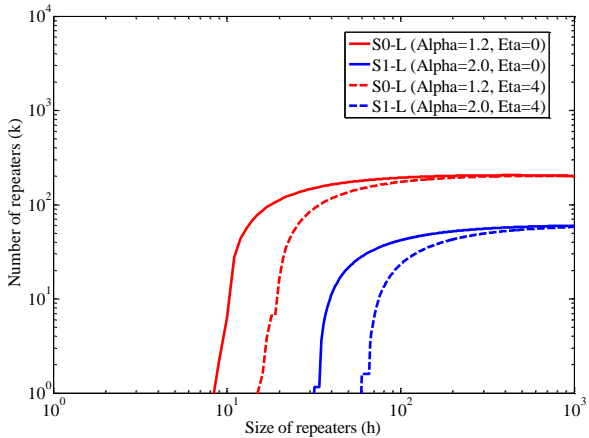
In the design phase of the flow, the design space exploration is performed as follows. Initially the solution boundaries that meet latency objectives for the two operating states, for the buses are found using (3-9). Recall that the associated boundaries are upper-bounds. Those bounds are shown by the red curve for s_0 (called s_{0-L}) and by the blue curve for s_1 (called s_{1-L}) in Figure 3-8 (a) for Bus #1, and in Figure 3-8 (b) for Bus #2. All the buses defined by the solutions inside the almost square regions defined by the red



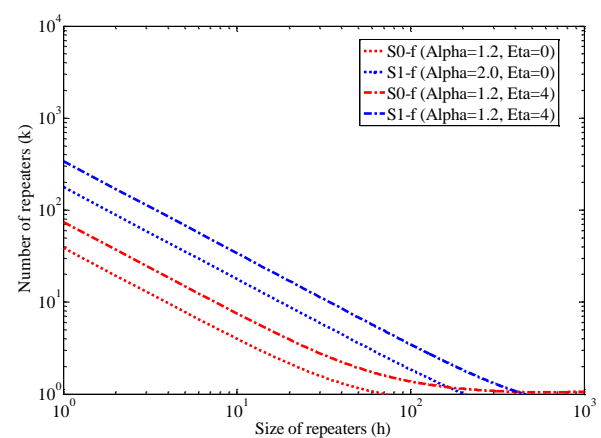
(a)



(c)



(b)



(d)

Figure 3-8: The design space exploration to meet the latency objectives in (a) Bus #1, (b) Bus #2; to meet the operating frequency objectives in (c) Bus #1, (d) Bus #2.

and blue boundaries meet the latency objective). The intersection of the two regions becomes the solution $\{(h_{PM-L}, k_{PM-L})\}$.

Later, (3-11) is employed to obtain the boundaries of the design space for operating frequency objectives. Those operating frequency related boundaries are the red dotted curve for s_0 (called s_{0-f}) and the blue dotted curve for s_1 (called s_{1-f}) as shown in Figure 3-8 (c) for Bus #1, and in Figure 3-8 (d) for Bus #2. Recalling that larger h and k

give higher operating frequencies, thus clearly the obtained boundaries are also bounds of the design space, with all solutions above each line meeting the respective frequency objective. The intersection of s_{0-f} and s_{1-f} defines $\{(h_{PM-f}, k_{PM-f})\}$.

In Figure 3-8, the design spaces have been explored based on the two boundary values of η . When wire/interconnects are considered to be fully shielded/isolated, η equals zero and in the case of existing fully aggressive adjacent lines, η may reach 4. According to Figure 3-8, we notice that the design space shrinks when the crosstalk effects increase ($\eta > 0$).

Before proposing the valid portion of the bus design space, based on the results of the above explorations, let's evaluate the behavior of ϑ in the given operating states. Figure 3-9 (a) and Figure 3-9 (b) depict the behavior of ϑ in the design space explorations of Bus #1 and Bus #2. According to this figure, ϑ increases with h ; accordingly, the design points with smallest h , located at the most left side in the design spaces, are the solution candidates to offer least energy and area overhead. Also, when $\eta=4$, almost twice larger ϑ_{PM} (hence twice larger bus overhead) is desired with respect to the case when $\eta=0$, to design the same buses. In other words, twice overhead is associated with the interconnect/bus design, if shielding is not properly applied.

Lastly, the intersection of $\{(h_{PM-L}, k_{PM-L})\}$ and $\{(h_{PM-f}, k_{PM-f})\}$ defines the valid portion of the design space $\{(h_{PM}, k_{PM})\}$ for the bus, meeting all design constraints shown in Figure 3-9 (c) and Figure 3-9 (d) for Bus #1 and Bus #2, as the region encompassing rigid horizontal lines, assuming k as an even integer number.

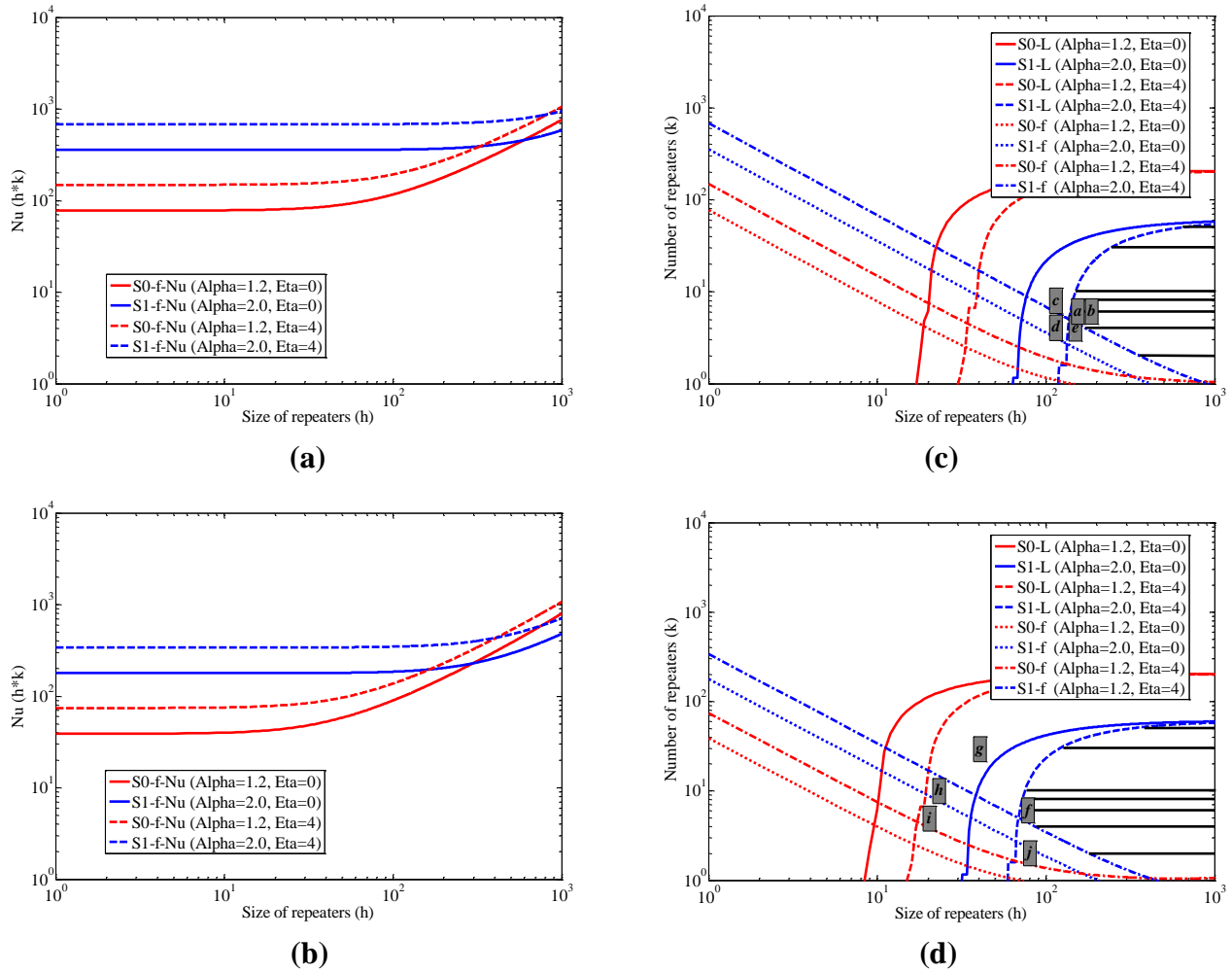


Figure 3-9: The required ϑ in the design space exploration of (a) Bus #1 and (b) Bus #2. The design spaces of (c) Bus #1 and (d) Bus #2.

Based on Figure 3-9 (c), the theoretical design solution of this case study becomes the design point (138, 4.9). This design point is the intersection of the latency and operating frequency design boundaries of s_1 . Nevertheless, the design space, which here is imposed by s_1 may vary, when the design requirements changes. Note that $k=4.9$, the number of repeater (stages), calculated with continuous analytic expressions, is not an integer. Imposing the constraint that k must be an even integer as no bus inversion is desired, the

feasible design solution with the smallest ϑ_{PM} becomes (168, 6). This solution is compared with four acceptable adjacent design points (**b**, **c**, **d**, and **e**), representing each of the four neighboring design regions, as shown by the design point **a** in Figure 3-9 (c), in order to highlight the efficiency of the proposed methods.

The same design exploration is performed for a bus with its line widths reduced to half. The analytical exploration analogous to the first scenario becomes the design point (69, 4.9) as depicted in Figure 3-9 (d). It is of interest that, the solutions are reduced to half, with respect to the first design case. The feasible design space solution with the smallest ϑ_{PM} , in this design case, becomes (84, 6) according to the proposed method, also as shown as the design point **f** in Figure 3-9 (d). This solution is compared with four acceptable neighboring design points (**g**, **h**, **i**, and **j**), representing four sampled design regions, as shown by Figure 3-9 (d), in order to highlight the efficiency of the proposed methods. Recall that the ten adopted design points are positioned as a dot and as the center of the corresponding characters in Figure 3-9 (c) and (d).

Two distinct signal transition patterns were applied to the buses: fully odd signal transition patterns (like “00000000” to “11111111”), representing the best design case and correlating to $\eta=0$ design variable, and, fully differential signal patterns (like “01010101” to “10101010”), representing the worst design case and correlating to $\eta=4$ design variable. The detailed HSPICE simulation results of the ten candidate design points are given in Table 3-4.

According to Table 3-4 and based on HSPICE simulation results, for the design of Bus #1, the design point **a** meets all design objectives with the least energy and area overhead.

Table 3-4: HSPICE simulation results of the two integrated bus structures (performance violations or sub-optimal results are underscored)

<i>Bus #</i>	1					2				
<i>Design Point</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
<i>h</i>	168	170	120	120	160	84	40	30	20	80
<i>k</i>	6	6	8	4	4	6	12	10	4	2
<i>ϑ</i>	1008	1020	960	480	640	504	480	300	80	160
<i>A</i> (μm^2)	274	<u>277</u>	261	130	174	137	130	81	21	43
<i>s_{0-L}</i> (ns), <i>Odd</i>	0.24	0.24	0.30	0.32	0.25	0.23	0.43	0.56	0.73	0.28
<i>s_{0-f}</i> (GHz), <i>Odd</i>	24.4	24.6	26.3	12.4	15.4	25.5	27.4	17.7	5.46	7.14
<i>s_{0-E}</i> (pJ), <i>Odd</i>	15.3	<u>15.4</u>	11.2	16.2	18.3	7.73	3.81	3.57	5.78	14.0
<i>s_{1-L}</i> (ns), <i>Odd</i>	5.75	5.70	8.02	7.30	5.67	5.84	<u>11.9</u>	<u>14.8</u>	<u>19.5</u>	5.30
<i>s_{1-f}</i> (MHz), <i>Odd</i>	1.04	1.05	0.99	0.54	0.70	1.02	1.00	0.67	<u>0.20</u>	<u>0.37</u>
<i>s_{1-E}</i> (pJ), <i>Odd</i>	1.51	<u>1.52</u>	1.11	1.81	1.96	0.76	0.37	0.38	0.77	1.69
<i>s_{0-L}</i> (ns), <i>Even</i>	0.39	0.39	0.53	0.54	0.45	0.39	0.73	1.00	1.48	0.55
<i>s_{0-f}</i> (GHz), <i>Even</i>	15.1	15.2	14.9	7.28	8.86	15.0	16.4	9.93	2.70	3.57
<i>s_{0-E}</i> (pJ), <i>Even</i>	21.1	<u>21.2</u>	15.5	24.5	26.6	10.5	5.19	5.24	10.3	22.8
<i>s_{1-L}</i> (ns), <i>Even</i>	10.0	9.95	<u>14.0</u>	<u>12.9</u>	10.0	10.0	<u>20.8</u>	<u>26.4</u>	<u>35.9</u>	9.53
<i>s_{1-f}</i> (MHz), <i>Even</i>	0.59	0.60	0.57	<u>0.30</u>	<u>0.39</u>	0.59	0.57	<u>0.37</u>	<u>0.11</u>	<u>0.20</u>
<i>s_{1-E}</i> (pJ), <i>Even</i>	2.37	<u>2.38</u>	1.75	3.10	3.24	1.19	0.59	0.64	1.41	2.97

Design point ***b*** meets all the design objectives but has higher energy and area overhead than the optimal design (design point ***a***). In this table, wherever a violation occurs (i.e. an objective is not met) or a result is sub-optimal, the associated result is underscored. Design point ***c*** does not meet the latency objectives when the signal patterns are fully differential, according to Figure 3-9 (c), the design point ***c*** is outside the design space defined by s_{1-L} when η equals 4. Design point ***d*** does not meet the latency or frequency objectives when the signal patterns are fully differential, according to Figure 3-9 (c), the design point ***d*** is outside the design spaces defined by s_{1-L} or s_{1-f} when η equals 4. Design point ***e*** does not meet the frequency objectives when the signal patterns are fully differential, according to Figure 3-9 (c), the design point ***e*** is outside the design space

defined by s_{1-f} when η equals 4. All these design points however meet the latency and performance objectives when the signal transitions are not fully differential.

A similar situation exists in the design of Bus #2. According to Table 3-4, and also validated by HSPICE simulations, the design point f meets all the design objectives with the least energy and area overhead, as considered to be the feasible near-to-optimum design point. Design point g does not meet the latency objectives s_1 regardless of the signal patterns; according to Figure 3-9 (d), the design point g is outside the design space defined by s_{1-L} . Design point h does not meet the latency objectives s_1 regardless of the signal patterns, nor the frequency objectives s_{1-f} when $\eta=4$, i.e. when signal patterns are fully differential. Design point i does not meet the latency or the frequency objectives s_1 regardless of the signal patterns; according to Figure 3-9 (d), the design point i is outside the design spaces defined by s_{1-L} or s_{1-f} . Design point j does not meet the frequency objectives s_{1-f} regardless of the signal transition patterns. According to Figure 3-9 (d), the design point j is outside the design spaces defined by s_{1-f} . All these design points on which Bus #2 is designed however, meet the latency and performance objectives s_0 .

According to Table 3-4, the reduction of the bus wire widths to half has resulted into the reduction of the energy and area overhead, for the same design requirements (for a correct comparison Bus #2 designed based on f should be compared to Bus #1 designed based on a).

Table 3-5: Extracted parameters for the management of integrated buses

Bus #	Des. (h, k)	PLD (%)	T_{0-f} (GHz)	T_{0-L} (ns)	T_{s-f} @ s_0	T_{s-L} @ s_0	T_{s-f} @ s_1	T_{s-L} @ s_1	$T_{s-limit}$
1	a (168, 6)	86	15.1	0.39	7.55	5.12	30.2	25.6	14.9
2	f (84, 6)	86	15.0	0.39	7.50	5.12	30.0	25.6	14.9

Table 3-6: HSPICE results for the two integrated buses subject to DVS (performance violations or sub-optimal results are underscored) M: Model

Bus #	Parameters	T_{s-f} @ s_0			T_{s-L} @ s_0			T_{s-f} @ s_1	T_{s-L} @ s_1
		$\alpha=1$	$\alpha=1.2$	$\alpha=2$	$\alpha=1$	$\alpha=1.2$	$\alpha=2$	$\alpha=2$	$\alpha=2$
1	V_{dd}^M	0.318	0.352	0.455	0.332	0.374	0.500	0.364	0.371
	L (ns)	<u>24.0</u>	<u>13.0</u>	<u>2.65</u>	<u>18.6</u>	<u>8.83</u>	1.54	<u>10.6</u>	9.37
	f (GHz)	<u>0.249</u>	<u>0.460</u>	2.262	<u>0.32</u>	<u>0.67</u>	3.89	0.56	0.63
	E (pJ)	1.48	1.81	3.10	1.60	2.06	3.78	1.94	2.02
	E Save (X)	9.88	8.07	4.83	9.07	7.14	4.21	1.20	1.16
2	V_{dd}^M	0.318	0.352	0.455	0.332	0.374	0.500	0.364	0.371
	L (ns)	<u>24.0</u>	<u>13.0</u>	<u>2.65</u>	<u>18.6</u>	<u>8.96</u>	1.64	<u>10.63</u>	9.35
	f (GHz)	<u>0.24</u>	<u>0.46</u>	2.25	<u>0.32</u>	<u>0.66</u>	3.64	<u>0.564</u>	0.641
	E (pJ)	0.74	0.91	1.54	0.80	1.03	1.92	0.97	1.01
	E Save (X)	9.88	8.07	4.83	9.07	7.14	4.21	1.20	1.16

As the control/management phase of the flow, the process is performed as follows. Initially, based on the obtained design solutions for the integrated Bus #1 as a , and for Bus #2 as f , the $PLDs$ as well as the nominal delays of the buses (here both the nominal bus operating frequency T_{0-f} and latency T_{0-L}), using (3-5) and (3-7) are calculated. Later, the scaling required at s_0 and s_1 for the operating frequency T_{s-f} and latency T_{s-L} , based on (3-5) and (3-7) are obtained. The limit of scaling for the buses, assuming a minimum defined supply voltage 400mV, and using (3-18) is further obtained. All the above steps for the given integrated buses are given in Table 3-5.

An important design note is that smaller scaling implies larger supply voltage; therefore in order for both the frequency and latency objectives be met during DVS, we must

perform the design exploration for supply voltage selection based on the smaller scaling rate, i.e. the worst case, here imposed by latency objectives as $T_{s-f} > T_{s-L}$. Another important fact with respect to Table 3-5 is, as the two considered buses have the same *PLD*, the acquired supply voltages for various operating states become equal. This is not the general case; if various bus lengths existed, various *PLDs* would be resulted and therefore various supply voltages would be required.

Later, the supply voltages for all the operating states need to be obtained. The related obtained solutions are given in Table 3-6. For the purpose of s_0 , using (3-20) and (3-21), the supply voltages based on $\alpha=1$ and $\alpha=2$ are obtained. Later, using (3-22) and in two iterations, the supply voltage based on $\alpha=1.2$ is calculated. The obtained supply voltage 352mV is in the range of 400mV which is coupled with $\alpha=2$ according to Table 3-3, therefore the acceptable design solution is the one based on $\alpha=2$, since α is a function of V_{dd} ; this fact has been ignored conventionally in modeling and was first highlighted in [104], also discussed in Section 3.2.2. In this case, the practical solution will be the one based on $\alpha=2$. For the purpose of s_1 , since the expected operating state supply voltage is 400mV, we only perform the design exploration based on $\alpha=2$. Since T_{s-f} and T_{s-L} are greater than $T_{s-limit}$, the minimum permitted value of 400mV could be assigned to V_{dd} ; assuming that smaller values are accepted (regardless of DC-DC converter capabilities, leakage issues, etc.), the obtained values could be also applied to V_{dd} .

According to Table 3-6, the obtained solutions based on T_{s-L} and $\alpha=2$, meet the target operating frequency and latency performance objectives at both s_0 and s_1 . Controlling the buses based on T_{s-f} or other values of α results into cases where the frequency or

latency objectives are not met in s_0 or s_1 (performance violations or sub-optimal results are underscored in the table). According to the proposed methods, at s_0 more than 4 times energy is saved compared to typical uniform DVS in power-managed systems, assuming the buses operate at the same supply voltage as logic. This efficiency is 16% at s_1 . Also note that, all the above reported results are obtained under worst case design conditions.

3.7 Chapter conclusions

In this chapter, a flow comprising methods for modeling, design and control (management) of interconnections in power-managed VLSI systems, were proposed. The proposed methods guarantee that the designed and controlled interconnects have minimum energy requirements while they meet their performance objectives at all the desired operating states.

The methods presented in this work could help designing and controlling energy-optimal interconnects, interconnect-centric components and/or mechanisms, particularly synchronization and communication mechanisms, etc. that need to meet desired performance objectives across various power-managed platform technologies: Microsystems enabled with generic run-time power-management, multiple voltage domains (voltage islands) and multiple clock domains systems, to name a few.

The contributions of this chapter are orthogonal to those from the other chapters; and if deployed simultaneously, the total net benefit may be the multiplication of the individual benefits.

Chapter 4

Design Exploration of Energy Managed Microsystems Based on Application

4.1 Chapter overview

In this chapter, models for Energy Performance (EP) estimation of energy-managed computing platforms are proposed. The energy models are based on the components of (embedded) application profile. The adopted method is inspired by Amdahl's law, which is driven by the fact that *'energy' is 'additive'; 'as time is additive'*. In this chapter, two classes of computing platforms have been considered. The difference between them is the ability to support power gating.

These models could be used for the design exploration of energy-managed (embedded) systems. The contributions of this chapter are orthogonal to those from the other chapters; and if deployed simultaneously, the total net benefit may be the multiplication of the individual benefits.

These models can guide a designer wishing to select between two classes of EM² platforms: one that offers an improved EP but requires a longer design-time and another

that offers fast prototyping but with less EP. The goal is assisting designers making decisions at the earliest design stage.

4.2 Foundations

Before proposing our models and analysis, the foundations of this research chapter are provided. The bases of this research are Amdahl's law and DVS.

4.2.1 Amdahl's Law

This work is inspired by Amdahl's law [115] and by its implications on EP models. In this section, we briefly review the basics of Amdahl's law and reformulate it to serve as the foundation of our performance models.

4.2.1.1 The Law

In the super-computing community, Amdahl's law has been widely accepted and utilized for Speed Performance (SP) estimation. It provides a simple, yet very useful method to estimate the potential acceleration in a parallelized computing platform. This method uses the result of application profiling, including the ratio of serial and parallel segments execution time, as well as the number of processing elements dedicated to the application program. In addition to multi-processor platforms, it has been widely utilized to determine total acceleration in uni-processor computing platform [116]. Equation (4-1) formulates Amdahl's law:

$$Speed\ Up = \frac{1}{s + \frac{p}{N}} \quad (4-1)$$

In this equation, the total workload is a function of normalized values; i.e. $(S+P) = 1$. S and P are typically the portions of the application that are sequential and parallel respectively, and N is the number of dedicated processing units (or processors), assuming that processing time of a part of an application is inversely proportional to the number of processing units.

4.2.1.2 Terminology Modification

In order to fully benefit from Amdahl's law, some reformulation is needed. P is traditionally referred to as the "parallel" portion of the execution time of a program. However, with some modern technologies, such as configurable computing platforms (including customizable, extensible, etc.), the computation speed is not necessarily improved only by dispatching parts of the application to an array of parallel processors. While acceleration can come from explicit parallelism (as in the traditional cases), it may also come from some form of implicit parallelism, cooperative enhancements (i.e. combination of multiple instructions/resources), pipelining, resource configurations, etc. Thus, the term "parallel" becomes restrictive. The term "*Enhanced*" (E) instead of "parallel" is used throughout this work. From this point of view, the term "*Resolute*" (R) is proposed to refer to the segment of the application that is not or cannot be enhanced (traditionally referred to as the "sequential" part of the application program). Also, N traditionally refers to the number of processors used, which is not the general case. To be more precise and due to the design capabilities in modern technologies, we refer to N as "*Kernel Acceleration*" (K), as this core acceleration may be obtained by different means, as discussed previously. In this case, we re-define "Speed Up" as the system SP , which

indicates the overall acceleration obtained by the whole system. Accordingly, (4-1) is reformulated as:

$$SP = \frac{1}{R + \frac{E}{K}} \quad (4-2)$$

4.2.2 Elements of DVS

DVS results in the reduction of system power/energy by means of changing system components operating voltage during run-time. Thus, DVS and its implications on logic delay, operating voltage, and power consumption, are reviewed here.

Accurate modeling of CMOS logic delay is complex when its nonlinear characteristics are considered. Yet, a simple, efficient, and reasonably accurate model was reported in [91] [92] [117]:

$$T = \frac{C_l * V_{dd}}{I_{dsat}} = \frac{C_l * V_{dd}}{\mu * C_{ox} * \frac{w}{l} * (V_{dd} - V_t)^\alpha} \quad (4-3)$$

In (4-3), C_l , V_{dd} and I_{dsat} are respectively the load-capacitance, the supply-voltage, and the drain-current in the saturation region; w and l are the effective width and length of the transistor, and finally μ , C_{ox} , V_t and α are its mobility, gate-oxide capacitance, threshold-voltage and saturation index respectively. Based on (4-3), we can equally consider:

$$T = \vartheta \frac{V_{dd}}{(V_{dd} - V_t)^\alpha} \quad (4-4)$$

In which ϑ is a factor combining all device parameters that are assumed to be constant in a given system. Now, if we consider the threshold-voltage as a fraction of the supply-voltage, i.e. $V_t = \kappa * V_{dd}$ where $0 < \kappa < 1$, (4-4) can be re-written as:

$$T = \vartheta \frac{V_{dd}}{[(1-\kappa)*V_{dd}]^\alpha} = \vartheta \frac{V_{dd}^{(1-\alpha)}}{(1-\kappa)^\alpha} \quad (4-5)$$

Similarly, (4-5) can be written as:

$$V_{dd} = \left[\vartheta^{\left(\frac{-1}{1-\alpha}\right)} * (1 - \kappa)^{\left(\frac{\alpha}{1-\alpha}\right)} * T^{\left(\frac{1}{1-\alpha}\right)} \right] \quad (4-6)$$

In VLSI, the dynamic power consumption is formulated as [92]:

$$P = \frac{C_l * V_{dd}^2}{T} \quad (4-7)$$

From (4-6) and (4-7), we conclude that:

$$P = \left[C_l * \vartheta^{\left(\frac{-2}{1-\alpha}\right)} * (1 - \kappa)^{\left(\frac{2\alpha}{1-\alpha}\right)} * T^{\left(\frac{1+\alpha}{1-\alpha}\right)} \right] \quad (4-8)$$

The value of α in typical full-swing super-threshold designs is 2 [91] [92]. In DSM technologies, the value of α decreases and converges toward 1 [92]. The typical values for κ vary according to the adopted technology. For a 90nm technology with $V_{dd} = 1V$ and $V_t=0.27V$, κ becomes 0.27. Now, if we assume $f = 1/T$ and $E = P * T$, for a typical design case in which $\alpha=2$; according to (4-6), there exists approximately a *linear* relationship between the frequency of operation and the supply-voltage. This linear relationship is widely seen in practice as well. Some commercial processors such as Intel XScale® [24] and Intel Pentium® [27] technologies confirm this linear characteristic. According to (4-8), a *cubic* relation exists between the power consumption and the frequency of operation. This partly stems from the *quadratic* relationship between the energy consumption and the supply voltage.

In this work, for simplicity, we define the constant β as:

$$\beta = \left(\frac{1+\alpha}{1-\alpha} \right) \quad (4-9)$$

Also as C_l, ϑ, κ and α have constant values in a given system, in this case we define the constant γ as:

$$\gamma = C_l * \vartheta^{\left(\frac{-2}{1-\alpha}\right)} * (1 - \kappa)^{\left(\frac{2\alpha}{1-\alpha}\right)} \quad (4-10)$$

Accordingly, (4-8) can be summarized by:

$$P = \gamma * T^\beta \quad (4-11)$$

Or by:

$$T = (P/\gamma)^{\frac{1}{\beta}} \quad (4-12)$$

The above transformation, from (4-11) to (4-12), will allow modeling energy performance based on the fundamental variables (such as the scaling rates, the application segments and the power-overhead) instead of using intermediate variables (such as the components of power-overhead and time due to scaling). We will use (4-12) later on in Section 4.5 for the purpose of energy performance modeling.

4.3 System power models

System energy models require detailed knowledge of system power models. In this section, system power models are proposed. This objective is met by means of introducing system models which are assumed to be aware of the application (software)

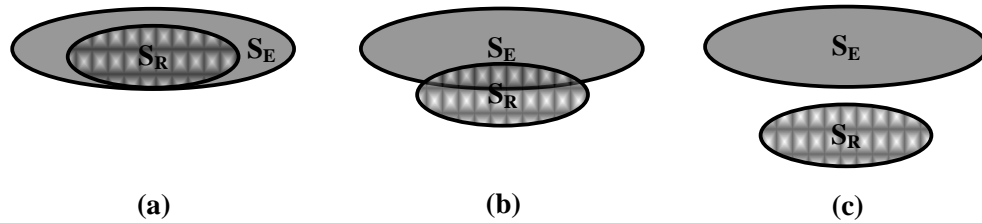


Figure 4-1: Modeling computing platforms based on their “resolute” and “enhanced” sets of processing elements.

segments. The proposed system power models will be later utilized to model system energy performance.

4.3.1 System models

Figure 4-1 shows, how system models defining two *sets* of processing elements in three different design settings would relate. In this figure, S_R is the set of processing elements which contribute to the processing of the *resolute* portion of an application program. This set correlates to the core of the computing platform. Then, S_E is the set of processing elements which contribute to the processing of the *enhanced* portion of the application program. This set correlates to the architectural enhancements, added to the core of the computing platform. These enhancements are a form of added extensions, in order to improve the processing efficiency of the computing platforms. Depending on the *application program, design technology, synthesis tools, compiler, type of enhancements, etc.*, the enhanced platform may be synthesized in one of the three design models (a)-(c). In (a), the architectural enhancements completely cover (embed) the core of the basic platform. This implies a complete architectural dependency between the two processing parts. In (b), the core and the enhancements have partial architectural dependency, and in (c), there is no architectural dependency between them.

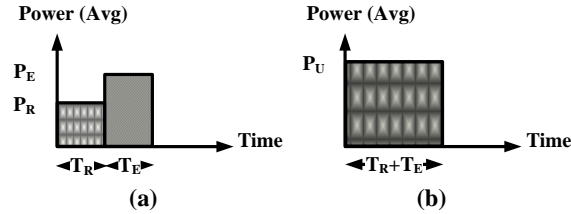


Figure 4-2: Power models of platforms: (a) with power gating, (b) without power gating.

An example of the resolute and enhanced processing elements can be a basic core of a processor, enhanced by a form of a co-processing engine. If the added co-processing engine is fully built on top of the basic engine and fully reuses the core, then the architectural dependency is complete. If the correlation between the co-processor and the basic core processing elements decreases, the architectural dependency among the two decreases accordingly.

4.3.2 System power models

Figure 4-2 shows a power-time graph illustrating the situation of two classes of platforms. The situation illustrated in (a) supports power gating, thus during the *resolute* processing time T_R , only the set of processing elements which contribute to the *resolute* processing segment (S_R) consumes power. On the other hand, during the *enhanced* processing time T_E , only the set of processing elements which contribute to the *enhanced* processing portion (S_E) consumes power. In technologies which do not support power gating however; as illustrated in (b), all processing elements consume power during the total processing time.

Accordingly, in platforms with power gating, the system power is defined by:

$$P_R = Power(S_R) \quad (4-13)$$

And:

$$P_E = Power(S_E) \quad (4-14)$$

In which P_R and P_E denote the average power consumption of the set of processing elements which contribute to the processing of the *resolute* and the *enhanced* portions of the applications program respectively.

Also, for the case of platforms without power gating, the power is modeled by:

$$P_U = Power(S_R \cup S_E) \quad (4-15)$$

If we expand the $S_R \cup S_E$ set based on S_R and S_E , we obtain:

$$P_U = Power(S_R) + Power(S_E) - Power(S_R \cap S_E) \quad (4-16)$$

In (4-16), $S_R \cap S_E$ is the subset of the processing elements consisting of the architectural dependencies between the *resolute* and the *enhanced* processing parts. These elements contribute to the process of both application portions/segments.

Now, if we assume that the power consumption of this common set is a fraction of the power consumption of the resolute processing elements (P_R), defined by a dependency factor d :

$$Power(S_R \cap S_E) = d * Power(S_R) = d * P_R \quad (4-17)$$

Based on (4-16) and (4-17), we conclude:

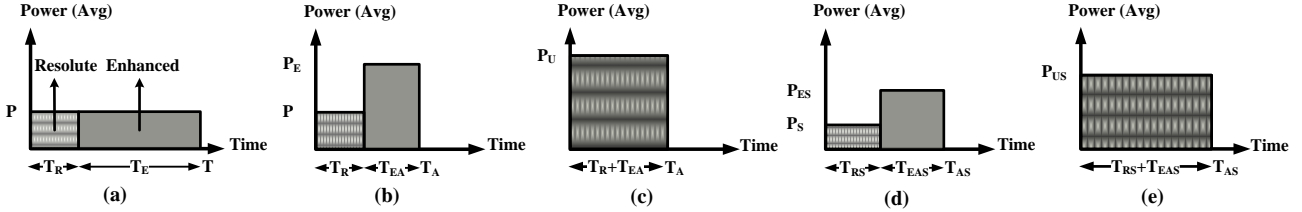


Figure 4-3: An application running on platforms: (a) the original case, (b) enhanced with power gating, (c) enhanced without power gating, (d) enhanced with power gating after DVS, (e) enhanced without power gating after DVS.

$$P_U = (1 - d) * P_R + P_E \quad (4-18)$$

In which the architectural dependency factor d is $0 \leq d \leq 1$. Obviously, *based on the application program, design technology, synthesis tools, type of enhancements, etc*, the dependency factor d may have different values in different design settings.

4.4 System energy performance models

The proposed energy performance models are inspired by Amdahl's law. The models consider the two architectural classes of computing platforms considered before. The first class comprises computing platforms capable of power gating along with (data-path) enhancements. This capability can result in very significant energy savings in VLSI systems. The other class comprises architectures with no power gating. Both classes are studied in terms of their energy performance, while (embedded) software is running on these platforms.

4.4.1 Platforms with power gating

In this class of computing platforms, while the core is processing, the enhanced/extended part is assumed to consume no power and vice-versa. In the following, we develop an energy performance model for this class of computing platforms.

In Figure 4-3(a), we consider an application program consisting of a resolute and an enhanced segment running on a computing platform. The platform is assumed to have an average power consumption P with a total processing time T (consisting of the resolute and the enhanced *processing times* T_R and T_E). In (b), architectural enhancements applied to the processor are assumed to result in a kernel acceleration K , thus reducing the processing time T_E to T_E/K (T_{EA}), with the overhead power P_E . Following these assumptions, we have:

$$EP_1 = \frac{\text{Energy Basic Platform}}{\text{Energy Enhanced Platform 1}} = \frac{P*T}{P*T_R + P_E*\frac{T_E}{K}} \quad (4-19)$$

EP_1 denotes the energy performance of the first class of computing platforms. Now, if we divide the power values by the core power P and *normalize* with respect to time T (substituting the total processing-time T by $T=T_R + T_E=1$), we obtain:

$$EP_1 = \frac{1}{\frac{T_R + P_o*\frac{(1-T_R)}{K*T}}{T}} \quad (4-20)$$

$\frac{T_R}{T}$ and $\frac{1-T_R}{T}$ represent R and E , the normalized times, in this case:

$$EP_1 = \frac{1}{R + P_o*\frac{E}{K}} \quad (4-21)$$

P_o in the above equations is the *normalized power-overhead* (i.e. the ratio of P_E/P). P_o could be obtained by means of available predictive power models (such as [59] [60] [61]) and/or power estimation tools (such as [118]) as it is *very architecture dependant and varies with technology*. Another important investigation worth performing is finding the upper-bound of energy performance in such computing platforms. In (4-21), if we consider the case where kernel acceleration is very large, knowing that E and P_o have *bounded* values (*which are typically correct assumptions for the energy-efficient utilization of computing platforms*), we obtain:

$$\lim_{K \rightarrow \infty} EP_1 = \lim_{K \rightarrow \infty} \left(\frac{1}{R + P_o \frac{E}{K}} \right) = \frac{1}{R} \quad (4-22)$$

Equation (4-22) reveals that *the achievable energy performance is bounded by the “resolute” portion of the software application*. This is a design concern similar to “net acceleration” (SP) being bounded by the sequential segment.

4.4.2 Platforms without power gating

In the first platform class, it was assumed that the core and the enhanced parts are dynamically power gated (by a power management unit). For the second class of computing platforms, it is now considered that this design option is not available. This may be due to technology limitations, system-level design tool limitations or restrictions associated to a given technology. Thus, in this platform class, it is assumed that only the functional modules (data-path and associated control) are enhanced to improve the computational acceleration, but the power distribution network is fixed and cannot be

gated. In this type of platforms, both the core and the enhanced part are consuming power at all times.

The components of the energy consumption, for this class of platforms, are illustrated in Figure 4-3(c). This figure illustrates the situation where the processor is enhanced to reach a kernel acceleration factor of K . Thus, T_E is reduced to T_E/K with the overhead power P_U . Therefore, the energy performance becomes:

$$EP_2 = \frac{\text{Energy Basic Platform}}{\text{Energy Enhanced Platform 2}} = \frac{P*T}{P_U*T_A} \quad (4-23)$$

Where T_A is the total time required for an enhanced application platform. Employing (4-18) results into:

$$EP_2 = \frac{P*T}{[(1-d)*P+P_E]*T_A} \quad (4-24)$$

If we divide the power values by the core power P , knowing that T/T_A denotes the achievable acceleration in the platform, i.e. the SP:

$$EP_2 = \frac{SP}{(1-d)+P_o} \quad (4-25)$$

In which P_o is the ratio of P_E/P . For this class of platforms, we again find the limit of achievable EP. According to (4-25):

$$\lim_{K \rightarrow \infty} EP_2 = \lim_{K \rightarrow \infty} \left[\frac{SP}{(1-d)+P_o} \right] \quad (4-26)$$

Combining (4-2) and (4-26) results into:

$$\lim_{K \rightarrow \infty} EP_2 = \lim_{K \rightarrow \infty} \left[\frac{1}{(1-d)+P_o} * \frac{1}{\left(R+\frac{E}{K}\right)} \right] \quad (4-27)$$

Assuming that E and P_o have bounded values for energy-efficient utilization in the computing platforms (which usually holds for such technologies), we have:

$$\lim_{K \rightarrow \infty} EP_2 = \frac{1}{[(1-d)+P_o]*R} \quad (4-28)$$

Equation (4-28) confirms that, the energy performance for this technology is also upper-bounded by the “resolute” segment of the application.

4.5 Energy performance models in systems subject to DVS

Following the previous discussion, in this section, models for the energy performance of the two former platform classes are proposed for the situation where they perform DVS.

4.5.1 Platforms with power gating

Let us consider an enhanced platform with power gating as in Figure 4-3(b). The platform has the core power P and processing time T_R for the resolute segment, and the enhanced processing power P_E with the processing time T_{EA} . In Figure 4-3(d), the processor is subjected to DVS with the scaling rates SR_R , SR_E and SR_S for the resolute segment, the enhanced segment and the overall system, resulting to the scaled processing times T_{RS} , T_{EAS} and T_{AS} . These considerations imply:

$$SR_R = \frac{T_{RS}}{T_R} \quad (4-29)$$

$$SR_E = \frac{T_{EAS}}{T_{EA}} \quad (4-30)$$

$$SR_S = \frac{T_{AS}}{T_A} \quad (4-31)$$

Note that we assume voltage may be downscaled, which implies that SR_R , SR_E and $SR_S \geq 1$. A scaling rate of 1 implies no scaling. Accordingly, as shown in Appendix I:

$$SR_S = A * \left(SR_R * R + SR_E * \frac{E}{K} \right) \quad (4-32)$$

Now, we explore the relation between the scaling rate and the power consumption for each processing part. For this, we utilize (4-12). Also, we consider dynamic power as a fraction ($1/\delta$) of the average power (dynamic power is the main portion of the average power, especially in high-performance systems operating in the super-threshold design region), and we recall from Section 4.2.2 that γ has a constant value. Based on these assumptions and according to (4-29), we have:

$$SR_R = \frac{T_{RS}}{T_R} = \frac{[(P_S/\delta)/\gamma]^{\frac{1}{\beta}}}{[(P/\delta)/\gamma]^{\frac{1}{\beta}}} = \left(\frac{P_S}{P} \right)^{\frac{1}{\beta}} \quad (4-33)$$

And, therefore:

$$\left(\frac{P_S}{P} \right) = SR_R^\beta \quad (4-34)$$

Similarly for the enhanced portion:

$$SR_E = \frac{T_{EAS}}{T_{EA}} = \frac{[(P_{ES}/\delta)/\gamma]^{\frac{1}{\beta}}}{[(P_E/\delta)/\gamma]^{\frac{1}{\beta}}} = \left(\frac{P_{ES}}{P_E} \right)^{\frac{1}{\beta}} \quad (4-35)$$

And:

$$\left(\frac{P_{ES}}{P_E}\right) = SR_E^\beta \quad (4-36)$$

Now, the Energy Performance due to voltage Scaling (EPS) for the first class of platforms, referring to Figure 4-3, is:

$$EPS_1 = \frac{E(b)}{E(d)} = \frac{[P * T_R + P_E * \frac{T_E}{K}]}{[P_S * T_{RS} + P_{ES} * T_{ES}]} \quad (4-37)$$

The division of (4-37) by the basic processor energy $E = P * T$, results into:

$$EPS_1 = \frac{\left[\frac{(P * T_R + P_E * T_E / K)}{P * T}\right]}{\left[\frac{(P_S * T_{RS} + P_{ES} * T_{ES})}{P * T}\right]} \quad (4-38)$$

With the expansion of the terms, we have:

$$EPS_1 = \frac{\left[\left(1 * \frac{T_R}{T}\right) + \left(P_O * \frac{T_E}{K * T}\right)\right]}{\left[\left(\frac{P_S}{P}\right) * \left(\frac{T_{RS} * T_R}{T}\right)\right] + \left[\left(\frac{P_{ES} * P_E}{P * P}\right) * \left(\frac{T_{ES} * T_E / K}{T}\right)\right]} \quad (4-39)$$

Applying (4-29), (4-31), (4-34) and (4-36) results into:

$$EPS_1 = \frac{R + P_O * \frac{E}{K}}{\left[(SR_R^\beta) * (SR_R * R)\right] + \left[(SR_E^\beta * P_O) * \left(R_E * \frac{E}{K}\right)\right]} \quad (4-40)$$

And finally:

$$EPS_1 = \frac{R + P_O * \frac{E}{K}}{SR_R^{(1+\beta)} * R + SR_E^{(1+\beta)} * P_O * \frac{E}{K}} \quad (4-41)$$

Now, we calculate the Total Energy Performance (TEP), comprising the energy performance due to enhancements and due to scaling. This would be obtained by:

$$TEP_1 = EP_1 * EPS_1 \quad (4-42)$$

The multiplication of (4-21) and (4-41), results into:

$$TEP_1 = \frac{1}{SR_R^{(1+\beta)*R} + SR_E^{(1+\beta)*P_o*\frac{E}{K}}} \quad (4-43)$$

The limit of total energy performance when $K \rightarrow \infty$, assuming P_o and the enhanced portion E are bounded values, becomes:

$$\lim_{K \rightarrow \infty} TEP_1 = \frac{1}{SR_R^{(1+\beta)*R}} \quad (4-44)$$

Equation (4-44) shows that the total energy performance is limited by the “*resolute*” portion of the software application program as well as by its “*scaling rate*”.

Another important design aspect to find is the optimum values for the scaling rates SR_R and SR_E , based on a given system scaling rate SR_S (which is determined according to the given processing time), in order to maximize the energy performance in such class of platforms. To find such optimum values, we apply the equivalent of SR_E to (4-43). Based on (4-32), the equivalent of SR_E is:

$$SR_E = \frac{SR_S - SR_R * A * R}{A * E / K} \quad (4-45)$$

Applying (4-45) to (4-43) results in:

$$TEP_1 = \frac{1}{SR_R^{(1+\beta)*R} + \left(\frac{SR_S - SR_R * A * R}{A * E / K} \right)^{(1+\beta)*P_o*\frac{E}{K}}} \quad (4-46)$$

In order to find the maximum of (4-46) for a given SR_S , we find the minimum of its denominator. Solving the derivative of the denominator of (4-46) for SR_R :

$$SR_{R,O} = e^{\left[\frac{\log \left[\frac{SR_S * (R * K + E)}{E + e^{\left[\frac{-1}{\beta} * \log \frac{1}{P_O} \right] * K * R}} \right] * \beta - \log \frac{1}{P_O}}{\beta} \right]} \quad (4-47)$$

Where $SR_{R,O}$ denotes the optimum value for SR_R . In this case, $SR_{E,O}$ could be obtained by means of employing (4-32) and based on the resulted $SR_{R,O}$. In other words:

$$SR_{E,O} = \frac{SR_S - SR_{R,O} * A * R}{A * E / K} \quad (4-48)$$

4.5.2 Platforms without power gating

Let us consider an enhanced platform without power gating as in Figure 4-3(c). The enhanced platform has a total system power consumption P_U and a total processing time T_A . In Figure 4-3(e), the enhanced processor is assumed to be subjected to DVS, with system scaling rate SR_S . The characteristics of such platform class imply that:

$$SR_S = SR_R = SR_E = \frac{T_{AS}}{T_A} \quad (4-49)$$

Using (4-12), and similar to (4-34) and (4-36), we obtain:

$$\left(\frac{P_{US}}{P_U} \right) = SR_S^\beta \quad (4-50)$$

In this case, EP due to scaling only, referring to Figure 4-3, becomes:

$$EPS_2 = \frac{E(c)}{E(e)} = \frac{P_U * T_A}{P_{US} * T_{AS}} \quad (4-51)$$

Applying (4-49) and (4-50) to (4-51) results in:

$$EPS_2 = \frac{1}{SR_S \beta * SR_S} \quad (4-52)$$

Or:

$$EPS_2 = \frac{1}{SR_S^{(1+\beta)}} \quad (4-53)$$

The TEP, comprising the energy performance due to enhancements and scaling, is obtained by:

$$TEP_2 = EP_2 * EPS_2 \quad (4-54)$$

This could be obtained by the multiplication of (4-25) and (4-53):

$$TEP_2 = \frac{A}{[(1-d)+P_o]*SR_S^{(1+\beta)}} \quad (4-55)$$

Or similarly:

$$TEP_2 = \left[\frac{1}{[(1-d)+P_o]*SR_S^{(1+\beta)}} * \frac{1}{\left(R+\frac{E}{K}\right)} \right] \quad (4-56)$$

The limit of TEP_2 when $K \rightarrow \infty$, assuming P_o and the enhanced portion E are bounded values, becomes:

$$\lim_{K \rightarrow \infty} TEP_2 = \frac{1}{[(1-d)+P_o]*SR_S^{(1+\beta)}*R} \quad (4-57)$$

Equation (4-57) shows that the total energy performance is limited by the “*resolute*” portion of the software application program as well as by its “*scaling rate*”.

4.6 Validation of the performance models

In this section, we explore the validity of the proposed performance models based on experimental results. For this purpose, the estimated energy performance of an extensible computing platform technology running three different classes of embedded applications has been compared with their actual implementations.

4.6.1 The ASIP platform technology and its design space exploration for energy performance

For the purpose of experimentations, the Tensilica Xtensa processor technology has been utilized [32]. Xtensa is an ASIP. ASIPs are a successful class of configurable platforms and are a popular architectural option to improve processing-efficiency, as discussed in Chapter 2. Some popular ASIP platforms (such as Tensilica Xtensa technology) yet, do not allow power gating. The Tensilica tools accept the definition of new instructions/enhancements (by means of the TIE language), and adding them to the Xtensa core in order to reach higher processing efficiency. Performance evaluation may be obtained by means of an instruction-set simulator and a profiling tool.

For this purpose, initially, the Tensilica Xtensa LX processor was extended by means of specific instruction-set(s), resulting into a kernel acceleration K (i.e. the speed improvement in the computational core), for three different classes of embedded applications. The design space of the system kernel acceleration (K) depends on several parameters, including the application critical segments that allow parallelism, the possible code and algorithmic optimizations of the application, available data dependencies in the

application, the number of data transfers in the application, etc. Each one of the adopted design threads in the available design space may result into a unique K , and an associated power overhead P_o . Such kernel acceleration usually is in the orders of magnitude, while the increase of the associated P_o is fairly small. Accordingly, based on the proposed models, large amount of energy will be saved by means of ASIP enhancements. Subject to DVS, such energy gains tend to increase even further.

For the purpose of design space exploration with respect to some target energy performance, the available set of design threads for K s and their associated P_o s can be applied to the energy performance models; if the potential energy gain is acceptable, the designer may continue with the same design thread, or change the thread to meet the energy objectives.

For the purpose of the implementations, the energy components of the extended and the basic processor core were compared to obtain the actual energy performance. For this, an FPGA port of the Xtensa LX processor from Tensilica is used. Simulations are performed to monitor the switching activity and to extract the power consumption values for each processor, with and without the defined instructions. The energy consumption is then deduced and comparisons are made (an in-depth discussion on the employed energy deduction method can be found in [119]). The conclusions of this work will be also valid for ASIC as the energy *ratios* (as opposed to absolute values) are studied.

The resolute and enhanced portions of the applications, i.e. R and E , were accordingly obtained by means of dividing the total number of cycle in the application belonging to that segment, by the total number of application cycles, according to the application

profile. For example the total number of application cycles that can be enhanced (here that can be accelerated) was divided by the total number of the application cycles. Similarly, the resolute portion was obtained by dividing the number of the application cycles that cannot be enhanced and is fixed –obtained from the application profile, over the total number of application cycles. Based on this technique, we can observe the complementary characteristic of the two segments as well.

In order to obtain power-overhead (P_o) values, initially a processor and its netlist (based on Xilinx NGO file) are generated using the Tensilica design flow. Then, a description of the specialized instructions can be attached to the Xtensa base core. At the end of the generation process, the NGO netlist of the processor is obtained, along with the Verilog description of the bus interface of the processor. The available design flow also provides tools to compile C programs and simulate them on the processor. From the generated NGO netlist and Verilog files, the Xilinx tool ngd2vhdl, produces a (gate-level) VHDL description of the FPGA configuration. Note that a similar process can be used when the target technology is an ASIC as only the specific tools differ. After compilation of the application, simulations are performed using ModelSim [120]. To characterize the activity inside the FPGA, the toggle analysis tool of the simulator is used. Statistics gathering begins after the reset sequence and ends when the main function of the C code reaches its end. The simulation result is a file with the name of each signal inside the FPGA associated with the number of transitions to 0 and to 1. Once the toggle rate of each signal is computed, the energy consumption of the processor can be deduced based on Xilinx power estimator WebTool [118]. The static power consumption of the logic

and the total power consumption of clock buffers are not considered as they differ between ASIC and FPGA technologies.

For the purpose of DVS, we considered a situation where the processing times are stretched to twice their initial values (i.e. $SR_S=2$). For this purpose the supply voltage of the Xilinx FPGA is set from its initial value 3V down to 1.5V. Such a design case may result in a quadruple increase in energy performance (according to (4-53) and (4-54)).

4.6.2 Case studies and implementation results

Three different applications have been employed as case studies: a *Motion Compensated Frame Rate Conversion* algorithm, a program that computes the *Fibonacci Series*, and the *Magnitude coder of the JPEG2000*. The Fibonacci Series is a *control-dominated* application, whereas the other two are *data-dominated* applications. Control-dominated applications typically involve light computations; thus by investing little hardware to enhance processing speed (here 1% overhead), a significant acceleration (here 7 folds) was reached. Data-dominated applications usually involve intense data computations, thus in order to achieve sufficient acceleration, large overhead may be induced to the system. In the following, we briefly introduce the software applications.

4.6.2.1 Motion Compensated Frame Rate Conversion

Motion Compensated Frame Rate Conversion (MC-FRC) algorithms [121] are used to multiply the frame rate of a video stream to match the requirements of the target application frame rate. Motion compensating methods use the motion inside the video to generate the interpolated frame. Consequently, a large effort is exerted to obtain motion estimation. In the present case, the algorithm used is based on block motion estimation.

In these algorithms, the video frames are divided into pixel blocks on which the motion is estimated. Such algorithms require lots of computation and high memory bandwidth. However, they follow a defined pattern and the computations are not data-dependent. Along with their highly parallelizable nature, they can be easily accelerated with application-specific instructions.

The extended instruction-set developed to accelerate MC-FRC was described in detail in [122]. It helps performing Block Motion Estimation, which is a compute intensive part of the algorithm. The proposed set of application-specific instructions exploits an additional wide register file to hold blocks of pixel values. These instructions compute from and manage data inside those registers according to the needs of Block Matching Algorithms (BMA) used to estimate the motion inside images.

Fourteen application-specific instructions created for the MC-FRC algorithm increase the processor power by 72% (P_o). Based on code inspection, such cooperative enhancements result into the equivalent of 188 basic core operations being performed concurrently (thus $K=188$). The speed performance (SP) reaches a factor of about 85.8 (the standard processor takes 61685 cycles to estimate the motion of one block whereas the extended processor takes only 719 cycles). The effective resolute segment portion of the application, (as defined/obtained by Amdahl's Law) based on the given K and A is 0.6% (R). Also, the obtained energy consumption ratio of the basic and the extended cores, obtained under the same respective assumptions (i.e. worst vs. worst, typical vs. typical, and best vs. best cases; explained in detail in [119]), is about 56 (EP_2). The total obtained energy performance after DVS with $SR_S=2$ is 224.

4.6.2.2 Fibonacci Series

The Fibonacci series is a simple, yet famous mathematical computation used in many benchmarking schemes [123]. It is defined by the following relation: $U_{n+1} = U_n + U_{n-1}$. The computations are sequential, and, except for a loop, they do not require any control. Only the first two numbers are needed as external values. This leads to absolutely no load instructions inside the main program loop. In our design, only the value of the n^{th} iteration is needed which implies no need for store instructions.

For this design case, only one instruction has been proposed. It allows performing 7 consecutive additions required to compute the corresponding numbers in a single instruction (K). The power-overhead is increased by only 1% (P_o). The speed performance (SP) reached is of about 6.55 (7014 cycles are needed to compute the first 7000 Fibonacci numbers, whereas the extended processor needs only 1071 cycles to do it). The effective resolute segment portion for this design case based on the given K and A , is 1.15% (R). The energy consumption ratio (basic vs. extended) under the same respective assumptions is about 12 (EP_2). The obtained energy performance after DVS (with $SR_S=2$) is 48.

4.6.2.3 Magnitude Coder of the JPEG2000

The magnitude coder is part of the Coefficient Bit Modeling (CBM) of the JPEG2000 standard [124]. The CBM precedes the MQ-coder which performs a context adaptive binary arithmetic coding method. Each bit is encoded according to its context. The magnitude coder is used before the MQ-coder to create the context-data pairs. Bit-plane coding (BPC) is applied on each bit-plane of the code-blocks to generate intermediate

data in the form of a context and a binary decision value. The magnitude coder is consequently working at the bit level. The computations are also heavily data-dependent with a lot of control (data and control dominated). This makes this algorithm very difficult to accelerate with application specific instructions.

In this design case, twelve application-specific instructions have been designed which extend the processor power by about 16% (P_o) and on average they result into 30 concurrent basic core processor computations (K), according to code inspection. The net-acceleration reached (A) is a factor of about 10.3 (the standard processor taking 16151 cycles to produce the bit-context associated with a 4 by 4 code block, whereas the extended processor only needs 1570 cycles to do so). The effective resolute portion for this design case is 6.6% (R). The energy consumption of the basic and extended ratio under the same respective assumptions is about 10.3 (EP_2). The total obtained energy performance after DVS with $SR_S=2$ is 41.2.

4.6.3 Comparing results

In order to model performance, the value of the architectural dependency factor d is needed. This dependency factor may have arbitrary values [125] [126]. Depending on various context specific factors (*application program, design technology, synthesis tools, compiler and its methods for optimizations, type of enhancements used, etc*), it may have different values. A typical dependency factor of 20% has been reported for data-dominated class of applications [125]. For our modeling, we accordingly swept d to extract its exact value and to explore the observable errors between the proposed models and the experimental results. Figure 4-4 depicts the error due to sweeping d for the given

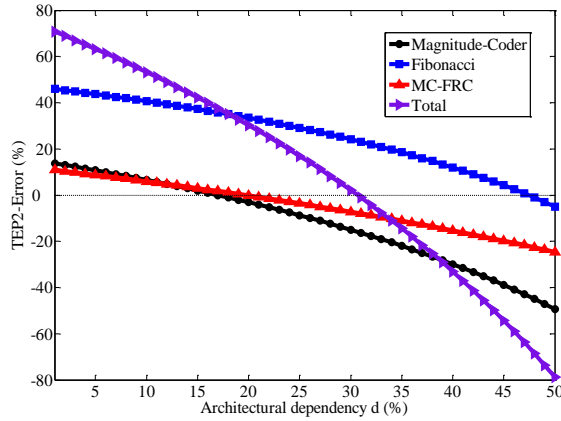


Figure 4-4: Sweeping the architectural dependency (d) in the three case studies, Error < 0 implies overestimating the energy performance.

case studies. We assume typical super-threshold operating region, thus $\alpha=2$ and $\beta=-3$.

This error is formulated by:

$$TEP_{2-Error} = \frac{TEP_2(Exp) - TEP_2(Model)}{TEP_2(Exp)} \quad (4-58)$$

According to this figure, MC-FRC, Fibonacci series and Magnitude-Coder have 18%, 46% and 15% architectural dependencies respectively. MC-FRC and Magnitude-Coder are data-dominated applications and their dependency factors are close to 20% as reported in [125]. The Fibonacci series is a control-dominated application and has higher architectural dependency. The average architectural dependency, including all classes of applications (that minimizes the error according to this figure) is 30%. In practice, such detailed architectural dependency profile may not be available initially, accordingly we consider two possible values of $d=20\%$ (representing data-dominated application class, and reported as a typical design value) and $d=30\%$ (representing all application classes) design cases, in our performance modeling.

Table 4-1: Energy performance in platforms for three embedded applications

	SP (Speed Prf.)	P_o (Power Ovr.)	K (Kernel Acc.)	R (Res. Portion)	EP_1 (Mod)	EP_2 (Mod, $d=0.2$)	EP_2 (Mod, $d=0.3$)	EP_2 (Exp)	EPR (Mod, $d=0.2$)	EPR (Mod, $d=0.3$)
MC-FRC	85.8	72%	188	0.6%	101.97	56.44	60.4	56	1.8	1.68
Fibonacci	6.55	1%	7	1.15%	77.44	8.08	9.2	12	9.5	8.39
Mag.-Coder	10.3	16%	30	6.6%	14.08	10.7	11.9	10.3	1.3	1.17

Table 4-2: Energy performance in enhanced platforms subject to DVS

	S_R	S_E	EPS_1 (Mod)	EPS_2 (Mod)	$EPSR$ (Mod)	TEP_1 (Mod)	TEP_2 (Mod, $d=0.2$)	TEP_2 (Mod, $d=0.3$)	TEP_2 (Exp.)	$TEPR$ (Mod, $d=0.2$)	$TEPR$ (Mod, $d=0.3$)
MC-FRC	2.10	2.02	4.28	4	1.07	437.1	225.7	241.6	224	1.9	1.8
Fibonacci	7.28	1.56	16.33	4	4.08	1265.3	32.3	36.9	48	39.1	34.2
Mag.-Coder	2.34	1.26	4.69	4	1.17	66.1	42.9	47.9	41.2	1.5	1.3

Table 4-3: TEP_2 -Error

	$d=20\%$	$d=30\%$
MC-FRC	-0.7%	-7.0%
Fibonacci	32.0%	23.0%
Mag.-Coder	-4.0%	-16.0%
Abs (Average)	9.1%	0%

Applying the EP models to the embedded application benchmarks produces the results in Table 4-1, Table 4-2, and Table 4-3. In Table 4-1, SP , P_o , K , and R respectively represent the obtained speed performance, the power overhead, the dedicated kernel acceleration, and the resolute segment portion associated with each application test case. EP_1 (Mod) and EP_2 (Mod) denote the estimated energy performance, utilizing the proposed performance models for the two classes of platforms. EP_2 (Exp) denotes the actual energy performance obtained using the implemented platform technology. EPR (Mod) denotes the estimated energy performance ratio of EP_1 (Mod) over EP_2 (Mod) respectively.

In Table 4-2, S_R and S_E denote the scaling rates of the resolute and the enhanced segments. $EPS_1 (Mod)$ and $EPS_2 (Mod)$ denote the estimated energy performance due to DVS only, based on the proposed performance models, and $EPSR (Mod)$ represents their ratio. $TEP_1 (Mod)$ and $TEP_2 (Mod)$ denote the total estimated energy performance including enhancements and DVS, utilizing the proposed performance models. $TEP_2 (Exp)$ denotes the actual total energy performance obtained using the implemented platform technology. $TEPR (Mod)$ denotes the total estimated energy performance ratio of $TEP_1 (Mod)$ over $TEP_2 (Mod)$ respectively.

In Table 4-3, the performance estimation errors, using (4-58), for various design cases and for the three embedded applications are reported. According to this table, for the two data-dominated applications, the typical dependency factor 20% provides accurate estimation of energy performances. The negative dependency factor implies an overestimation of energy performance reported by the models. According to this table, for the control-dominated class of applications, a larger dependency is expected. According to this table, using a typical dependency factor 20%, the proposed models show 9.1% error on average, compared to their actual implementation results for all classes of embedded applications. According to the results (validated by Figure 4-4), the typical dependency factor 30% maximizes the accuracy of the proposed models, considering all the classes of embedded applications. A good rule of thumb would be to use 20% dependency factor for data dominated applications, a larger value -up to 50%-, for control dominated applications. Therefore, a dependency factor of 30% may be used to represent a typical application comprising both control and data dominated segments.

Taking the results of Table 4-3 into consideration, according to Table 4-1, large energy performance gains may be obtained by power gating, according to *EPR*. The relative energy performance may increase up to 80% for the data-dominated class of applications (1.8 for MC-FRC, considering $d=20\%$) and may increase up to 8 folds for control-dominated class of applications (8.39 for Fibonacci, considering $d=30\%$). Based on the results in Table 4-2, power gating can be very effective in platforms subject to DVS, according to *TEPR*. As reported in this table, the relative energy performance may increase up to 90% for the data-dominated class of applications (1.93 for MC-FRC, considering $d=20\%$) and may increase more than 34 fold for control-dominated class of applications (34.2 for Fibonacci, considering $d=30\%$). These results reveal that the use of power gating can be a vital solution for systems with very limited energy resources (especially for the case of control-dominated class of applications). This observation is significant when considering that some modern popular platforms such as Xtensa that do not support power gating.

4.7 Chapter conclusions

In this chapter, models for energy performance estimation of energy-managed computing systems were proposed. The energy models are based on the components of (embedded) application profile. These models could be used for design exploration of green energy-managed (embedded) systems. The contributions of this chapter are orthogonal to those from the other chapters; and if deployed simultaneously, the total net benefit may be up to the multiplication of the individual benefits.

Chapter 5

Design Exploration of Energy Managed Microsystems Based on Activity

5.1 Chapter overview

In this chapter, models to estimate the system energy according to its “activity” are proposed. By “activity” we mean the frequency or rate, at which the computing platform, i.e. the system, performs a set of predefined application functions. We evaluate the activity by Calls Per Second (CPS) or Hertz (Hz) [127]. Good estimations of energy requirements are very useful when managing the system activity in order to extend the battery lifetime.

In order to validate the proposed models, a generic low duty-cycle ZigBee[®] Wireless Sensor Network (WSN) application has been considered as a case study. Experiments confirm the accuracy of the estimated energy consumption values.

These models could be used for the design exploration of (portable) embedded EM². The contributions of this chapter belong to the system/application design abstraction level, and are orthogonal to those from the other chapters; if deployed simultaneously, the total net benefit will be much greater.

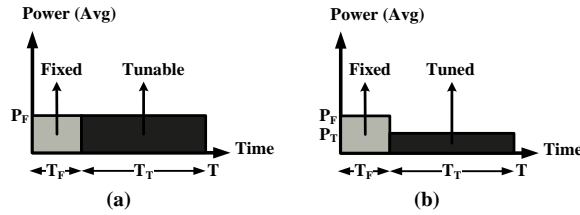


Figure 5-1: Power/Energy profile of an application, including “fixed” and “tunable” segments, running on an embedded system.

5.2 Energy estimation of embedded systems based on their activity

In this section a generic application-driven energy model based on the notion of activity is first proposed. Later, a specific platform-driven energy model for a ZigBee[®] WSN platform technology is introduced.

5.2.1 A generic application-driven energy performance model based on the notion of system activity

Application profiling is a well known method for speed and energy performance exploration in computing systems, as discussed in detail in Chapters 2 and 4. This method considers the fixed and tunable application segments; which can be adjusted to improve the speed and/or energy efficiencies.

Figure 5-1 (a) shows an embedded system, processing an application program composed of fixed and tunable segments. The platform is assumed to have an average power consumption P_F , representing the power overhead of the system with a *fixed activity*, and a total processing time T , consisting respectively of the fixed and the tuned processing

times T_F and T_T . In (b), application and/or architectural tunings are applied, which results in an additional processing state with power consumption P_T , representing the *tuned activity* during the time interval T_T . Following these assumptions:

$$EP = \frac{\text{Energy Fixed Platform}}{\text{Energy Tuned Platform}} = \frac{P_F * T}{P_F * T_F + P_T * T_T} \quad (5-1)$$

EP denotes the Energy Performance of the tuned embedded computing platform. Now, if we divide the power values by the basic power overhead P_F , and *normalize* with respect to time T (substituting T by $T=T_F + T_T=1$), we obtain:

$$EP = \frac{1}{\frac{T_F}{T} + P_o * \frac{(1-T_F)}{T}} \quad (5-2)$$

Let $\lambda_F = \frac{T_F}{T}$ and $\lambda_T = \frac{1-T_F}{T}$ represent the fixed and tuned application *portions* ($\lambda_F + \lambda_T=1$), respectively. In this case:

$$EP = \frac{1}{\lambda_F + P_o * \lambda_T} \quad (5-3)$$

P_o in the above equations is the normalized power-overhead, i.e. the ratio of P_T/P_F , the average power ratio of the two processing states with distinct activities. P_o can be obtained by means of predictive power models, such as the one given in the next section, power estimation tools, etc. as it is very application/architecture dependent and varies with technology.

Another important investigation worth performing is finding the upper-bound of energy performance in such tunable platforms. In (5-3), if we consider the case where the system

is totally shut-down during the tunable segments (i.e. absolutely no power consumption, thus $P_T=0$) during T_T ; we obtain:

$$\lim_{P_T \rightarrow 0} EP = \lim_{P_T \rightarrow 0} \left(\frac{1}{\lambda_F + P_T \lambda_T} \right) = \frac{1}{\lambda_F} \quad (5-4)$$

Equation (5-4) confirms that the achievable energy performance considering the system activity is bounded by the portion of the application with “fixed” activity, i.e. λ_F .

5.3 Energy performance model for generic ZigBee[®] application platforms based on the notion of activity

In this section, we propose an energy performance model for ZigBee[®] applications. The proposed energy model is based on the Texas Instruments ZigBee[®] compliant protocol implementation called the Z-Stack [128], as well as a generic control/monitoring application [129].

Let us first consider a generic energy model for a wireless system expressed as:

$$E = E_F + E_{Rx} + E_{Tx} + E_S \quad (5-5)$$

In (5-5), E_F is the energy consumption for the platform, performing required local processing functions. E_{Rx} and E_{Tx} are respectively the energy consumption when receiving and transmitting; and E_S is the energy consumption during sleep (low power) mode, assuming power-management features are enabled such that when the system has no function to perform, it enters the sleep (low power) mode.


```

In every second ( )
{
    At 100 millisecond time stamps ( )
    {
        Perform the application, based on the activity factor.
        If no activity left, transmit status once.
        If the status is transmitted, listen.
    }
    Go to sleep (low power) mode.
}

```

Figure 5-2: Pseudo code of a generic control monitoring application implemented over TI Z-Stack.

Now consider the pseudo code of a generic periodic control/monitoring application implemented using the TI ZigBee® Z-Stack in a WSN with a star topology (see **Figure 5-2**) available in [129]. Based on this code, a generic energy model for a ZigBee® system application activity factor α_s and processing period T may be expressed as:

$$E_Z(\alpha_s) = \alpha_s P_F T_F + \beta P_{Rx} T_{Rx} + \gamma P_{Tx} T_{Tx} + P_S T_S \quad (5-6)$$

Where

$$T_S = T - (\alpha_s T_F + \beta T_{Rx} + \gamma T_{Tx})$$

$$\begin{cases} \gamma = \beta = 0 & \alpha_s \geq N \\ \gamma = 1, \beta = N - (\gamma + \alpha_s) & \alpha_s < N \end{cases}$$

In (5-6) α_s, β and γ are activity factors that have non-negative integer values. In this equation, α_s is the system application activity factor for the application functions, β is the activity factor for the number of ZigBee® protocol receptions and γ is the activity factor for the number of ZigBee® protocol transmissions. Activity factors β and γ are dependent on α_s , as expressed in (5-6). N denotes the nominal (and not the maximum) available

slots for application activity. Based on **Figure 5-2**, in the application with the time period 1s and sampling time stamps 100ms, N becomes 10. In the above formulation, P_F and T_F refer to the processing power and time required to perform application function(s). Also, P_{Tx} and T_{Tx} , P_{Rx} and T_{Rx} , and P_S and T_S refer to the processing power and times of the platform during transmission, reception and sleep mode respectively.

In order to obtain an energy-performance model, we must consider the average power consumption $P_Z(\alpha_s)$. This can be deduced from $E_Z(\alpha_s)/T$. In this case the energy performance for an activity managed ZigBee[®] wireless embedded system, performing a *periodic* application, becomes:

$$EP_Z(\alpha_s) = \frac{1}{\lambda_F + P_{oz} * \lambda_T} \quad (5-7)$$

Where

$$P_{oz} = \frac{E_Z(\alpha_{s-T})}{E_Z(\alpha_{s-F})}$$

Equation (5-7) can be used to explore the energy performance of generic ZigBee[®] applications. In (5-7), $E_Z(\alpha_{s-T})$ and $E_Z(\alpha_{s-F})$ represent the energy consumption of the platform, processing the tuned and fixed application segments with the system activity factors α_{s-T} and α_{s-F} , respectively.

5.4 Experimental results

In this section, we evaluate the proposed energy models applied to the prediction of the energy consumed by the TI ZigBee[®] WSN technology platform that is based on the CC2530 SoC [130]. The SoC executes both the processing required by the application as

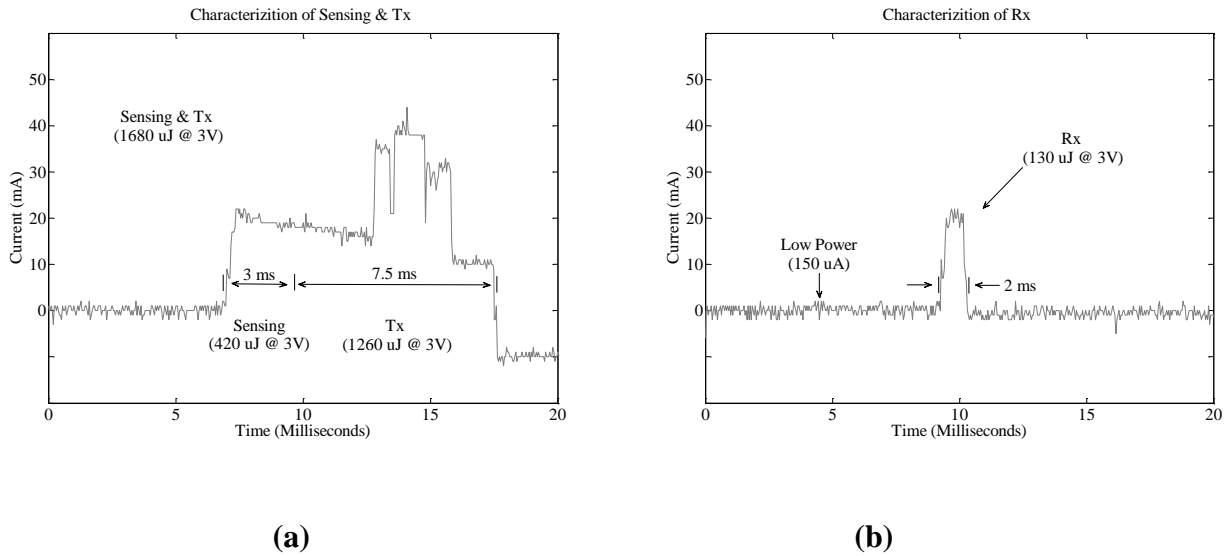


Figure 5-3: Experimental characterization of power/energy profiles of a ZigBee® WSN platform during (a) sensing, and, data transmission, (b) data reception.

well as network communications. This network characterized by a star topology is considered for its simplicity. A temperature sensing application representing a generic control/monitoring application has been used [129]. A 3V battery source was used in the experiments. Instantaneous current measurements (and their accumulation) have been performed to deduce the equivalent average power and energy values of the application platform.

In order to evaluate the energy models, we should first characterize the power/energy profile of these platforms during computations and communications. **Figure 5-3** reports typical measurements. According to **Figure 5-3(a)**, the embedded application, in each activity period, performs a sensing operation (local sensing and related processing), as well as transmission of the measured values over the network. When no sensing is performed, only some control information is transmitted (or received) over the network (piggybacked). In **Figure 5-3(b)**, data reception is characterized (listening mode).

Table 5-1: Power/energy profile of the TI ZigBee[®] WSN platform (Vdd=3V)

State	Avg. Power (mW)	Duration (mS)	Energy (uJ)
Sensing	140	3	420
Tx	168	7.5	1260
Rx	65	2	130
Sense & Tx	160	10.5	1680
Sleep	0.45	-	-

According to the application, when the platform is idle, it enters its low power (sleep) mode, as can be seen from **Figure 5-3(a)** and **3(b)**. Table 5-1 quantifies the power/energy profile related to **Figure 5-3**. Note that the energy profile parameters given in this table are typical value. These values may significantly vary with packet length and channel characteristics. Such uncertainty may result into significant modeling errors.

After characterizing the power/energy values, the activity of the application is tuned in different states, and the measured energy is compared with the one predicted by the energy model (5-6). Figure 5-4 reports the measured current consumption of the platform when operating with four different application profiles, each profile having a different activity factor α_s . Figure 5-4(a) demonstrates the case where the platform has zero activity. As can be seen in this figure, even when the platform has no functional activity, it performs communication protocol related functions and consumes some energy. In this figure, no sense-and-transmission ($\alpha_s=0$) action is performed, but one transmission action ($\gamma=1$) is performed.

Note the clear 100ms time stamp structure. There are 9 time stamps where the platform is in the listening mode ($\beta =9$). Figure 5-4(b) reports measured current when the platform

has α equal to 1. In this case, a sense-and-transmission ($\alpha_s=1$) as well as one protocol transmission ($\gamma=1$) are performed and in other 100ms time stamps, the platform is in listening mode ($\beta =8$).

In Figure 5-4(c), the platform operates with α_s equal to 2. In this case, two sense-and-transmission activities ($\alpha_s=2$) and one protocol transmission ($\gamma=1$) are performed and in other 100ms time stamps, the platform is listening ($\beta =7$). In Figure 5-4(d), the platform

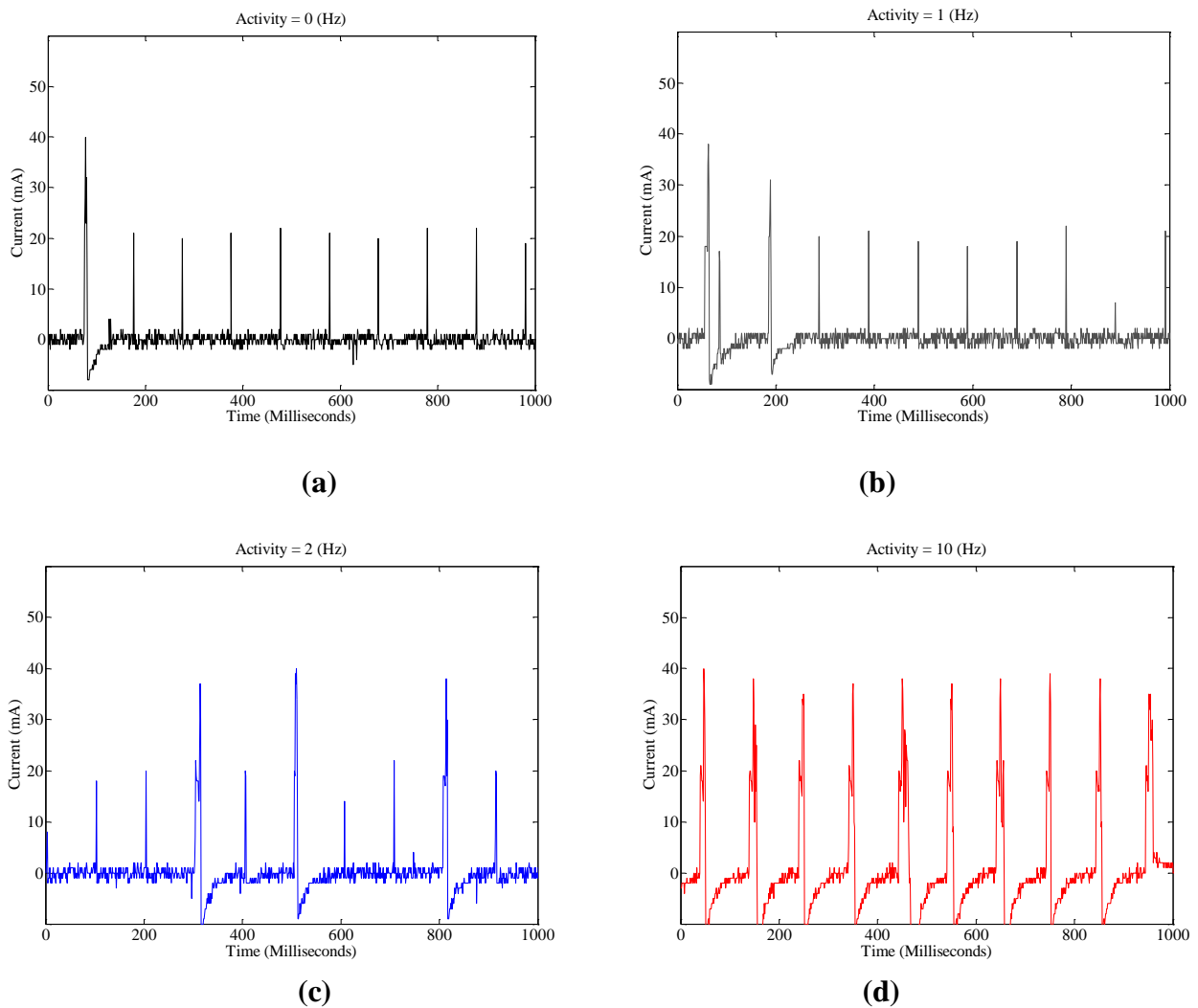


Figure 5-4: Charactering WSN application platform in four different operating states (a) $\alpha_s = 0$, (b) $\alpha_s = 1$, (c) $\alpha_s = 2$, (d) $\alpha_s = 10$.

Table 5-2: Characterizing the TI ZigBee® WSN platform with four static activity states (energy unit is in uJ)

State	α_s	β	γ	E (Mod)	E (Exp)	Error
1	0	9	1	2869	3500	18%
2	1	8	1	4415	4570	3%
3	2	7	1	5961	6030	1%
4	10	0	0	17202	17300	1%

is active in every 100ms time stamps ($\alpha=10$) and $\gamma=\beta=0$.

Table 5-2 summarizes the values in Figure 5-4. According to this table, the proposed model, in (5-6), is fairly accurate for energy modeling of generic ZigBee® applications when configured with suitable parameters. According to this table, when the activity factor increases, the error decreases. The error is $\leq 3\%$ when the system has some activity, i.e. $\alpha_s \geq 1$.

In the previous experiments, the activities of the platform were statically chosen during the whole operating period. In the next experiment we changed the activity for some portions of the operating period. This was done by means of employing the internal timers of the platform to dynamically change the platform activity.

Table 5-3 quantifies the obtained energy performance from the platform and those obtained from (5-7). According to this table, in the design case 1, the platform operates with activity rate 10 (Hz) 50% of the time, the rest of the time, the activity rate is reduced to 1 (Hz). For design case 2, the low activity rate is set to 0 (Hz). In both of these design cases, since λ_F equals 0.5, the upper-bound of the energy savings factor is 2, based on (5-4). This means that even if during 50% of the time, the platform is completely shut down, instead of having some activity (note that shut down is different from activity rate equal

Table 5-3: Characterizing the TI ZigBee[®] WSN platform in various cases with dynamic activities

Case	λ_F	λ_T	α_{S-F}	α_{S-T}	P_{oz} (Mod)	EP (Mod)	EP (Exp)	Error
1	0.5	0.5	10	1	0.25	1.60	1.58	1%
2	0.5	0.5	10	0	0.16	1.72	1.66	3%
3	0.25	0.75	10	1	0.25	2.28	2.24	1%
4	0.25	0.75	10	0	0.16	2.70	2.50	8%

to zero, as the platform still consumes some energy due to the communication protocol), the obtained energy savings is limited by a factor of 2. In design case 3, for 25% of the time, the platform operates with high activity rate 10 (Hz), whereas for 75% of the time, the activity rate is reduced to 1 (Hz). For design case 4, the low activity rate is set to 0 (Hz). In both cases, the upper-bound of the energy savings factor is 4. The design cases 1 and 2, with λ_F equal to 50%, may represent a situation where a system is monitoring an area or a biological entity during day-time with a relatively high activity and during night-time with a low activity. However, design cases 3 and 4, with λ_F equal to 25% may represent the case of a system monitoring an area or a biological entity during a specific season in the year with high activity, and the other 3 seasons with low activity. According to **Table 5-3**, the proposed energy performance model is fairly accurate with an average error of 3%, over estimated energy performance.

5.5 Chapter conclusions

Many energy-restricted embedded systems have fairly small energy budget coming from batteries. For this purpose, static and possibly dynamic optimizations need to be applied

to both the hardware and software design abstraction layers, in order to improve system efficiency.

In this chapter, models for estimating energy requirements based on activity were proposed. By activity, we mean the rate at which the computing platform performs a set of predefined application functions. The proposed models can be used for both static and/or dynamic (on the fly) design exploration and management of energy managed embedded technologies. In order to validate the models, a generic low duty-cycle ZigBee[®] WSN control and monitoring application was considered as a case study.

These models could be used for design exploration and management of green energy-managed (embedded) systems. The contributions of this chapter are orthogonal to those from the other chapters; and if deployed simultaneously, the total net benefit may be the multiplication of the individual benefits.

Chapter 6

Conclusions

6.1 Summary

Aggressive power/energy reduction is one of the significant challenges that all segments of semiconductor industry have encountered in the past few years. This challenge needs to be met, while on the other hand, environmental awareness and designing “green electronics” has become an additional driver for (ultra) low energy design of microelectronic systems.

Supervisory system design is introduced as a promising trend for VLSI system realization, to address such technological challenges. Dynamic energy management, a successful class of supervisory design scheme, is one of the unique solutions that can address the simultaneous challenge of high-performance, (ultra) low-energy and greenness in many classes of computing systems, including high-performance, embedded and wireless.

Consequently, the focus of this thesis was toward a holistic approach for efficient realization of EM² -Energy Managed Microsystems-, with the aim of maximizing their energy-efficiency and/or operational lifetime. The proposed solutions given in this thesis are applicable to many classes of computing systems, including high-performance and mobile computing systems. They can contribute to make these technologies “greener”.

The proposed solutions are multilayer since they belong to, and some applicable to, multiple design abstraction layers. If the solutions are deployed to the same modules, in a vertical system integration approach, the total benefit can be as large as the multiplication of the individual benefits. If they are deployed to various independent components, the net benefit will be the accumulation of the individual benefits.

At high-level, the premise of this thesis was toward guiding system designers to

- Design and manage EM^2 based on system interconnection,
- Design and manage EM^2 based on system application,
- Design and manage EM^2 based on system activity.

In the following sections, the review of major contributions and the possible future work of this thesis are given.

6.2 Review of the thesis contributions

This thesis initially focused on the modeling, design and management of EM^2 interconnections. In DSM, as VLSI systems become interconnect-centric, accurate modeling and design of interconnections become very critical for efficient realization of such systems. For this purpose, a design flow was proposed. It comprises methods for modeling, design and management of EM^2 on-chip interconnections. This flow guaranteed that the designed interconnects have *minimum* energy requirements, and that *they meet all the performance objectives, in all the system operating states within system specification*. The proposed flow showed superior performance improvements with

respect to its modeling, design and control of EM² interconnections. The results of this part of the research were reported in [104] [131] [132] [133] [134].

Later, models for energy estimation of EM² were proposed. Modeling and design explorations of EM² for energy performance are valuable stepping stones that can help accelerate the design process. These models are based on the components of application profile. The adopted method was inspired by Amdahl's law, which was driven by the fact that 'energy' is 'additive'; 'as time is additive'. These models can be used for the design space exploration of EM². The proposed models are high-level, and showed fair accuracy, *9.1% error on average*, when targeting various embedded benchmarks. The results of this part of research were reported in [79] [135].

Finally, models to estimate the EM² energy according to its "activity" requirement were proposed. By "activity" we mean the rate at which EM² performs a set of predefined application functions. Good estimations of energy requirements are very useful when designing and managing EM² activity, in order to extend their battery lifetime. The study of the proposed models on some ZigBee[®] Wireless Sensor Network (WSN) application benchmark confirmed a fair accuracy, *3% error on average*, for the energy estimation models. The results of this part of research were reported in [136].

6.3 Recommendations for future work

Some possible future works to extend the proposed work of this thesis are explained as follows.

6.3.1 Future works in the domain of EM² interconnection realization

In the domain of EM² interconnection realization, taking the signal slew-rate as well as the interconnection inductance effects into account, can further improve the accuracy of the proposed performance models given in Chapter 3.

Adaptive Body Biasing (ABB), on the other hand, has been introduced as an effective method that may be used to improve the energy consumption in the CAS. Performing simultaneous DVS and ABB can be very effective for ULP. Accordingly the proposed repeater insertion modeling and design methods, given in Chapter 3, can be modified to support EM² that support ABB as well.

In our interconnection realization in Chapter 3, we used conventional buffers, and considered the conventional method of signaling, i.e. full-swing single-mode. The proposed methods can be modified to support various types of logic families for buffers, including Dual-Vt, Multi-Vt, Dynamic-Vt, as well as various types of signaling schemes, including low-swing, current-mode and differential. Based on these researches, we could have complete profile of EM² that support various logic families and signaling schemes.

Analysis of Via, Process Voltage Temperature (PVT) variations, soft-error, reliability, yield, and temperature effects in 2D and 3D EM² interconnection realizations, open another wide set of research areas to explore.

6.3.2 Future works in the domain of application-level EM² realization

In the domain of application-level realization of portable EM², battery-awareness is a criterion that can effectively contribute to system energy-efficiency. Modifying the

proposed application-driven and activity-driven energy performance models, given in Chapters 4 and 5 respectively, to support battery-awareness characteristic, can improve the impact of the proposed solutions by several folds.

In the case of wireless portable EM², the characterization and the analyses of per bit energy operation, and per bit energy transmission (for communication), and their tradeoffs, are very critical and valuable. Modifying the proposed energy performance models given in Chapter 5, to consider the processing-communication tradeoffs with the support of battery-awareness characteristic can tremendously add to the value of our modeling and design exploration work.

Appendix I

Recalling from Chapter 4, in order to find the relation between the systems scaling factor SR_S , and the applications segments scaling factors SR_R and SR_E , referring to Figure 4-3(b) and 3(d), based on (4-31) we have:

$$SR_S = \frac{T_{RS} + T_{EAS}}{T_R + T_{EA}} \quad (I-1)$$

Now by applying (4-29) and (4-30), as well as substituting T_A by its equivalent, we have:

$$SR_S = \frac{(SR_R * T_R) + (SR_E * T_E / K)}{T/A} \quad (I-2)$$

Or:

$$SR_S = A * \left(SR_R * \frac{T_R}{T} + SR_E * \frac{T_E}{K * T} \right) \quad (I-3)$$

T_R/T and T_E/T are the definitions of the resolute and the enhanced portions according to Amdahl's law; therefore:

$$SR_S = A * \left(SR_R * R + SR_E * \frac{E}{K} \right) \quad (I-4)$$

Bibliography

- [1] Power (physics). Available: Wikipedia.org/Power
- [2] S. Borkar, “Design challenges of technology scaling”, *IEEE Micro*, vol. 19, pp. 23–29, July–Aug. 1999.
- [3] Jan Rabaey, *Low Power Design Essentials*. Springer, 370 pages, 2009.
- [4] International Technology Roadmap for Semiconductors, 2007. Available: www.itrs.net.
- [5] San Murugesan, “Harnessing Green IT: Principles and Practices”, *IT Professional*, vol. 10, no. 1, pp. 24-33, Jan 2008.
- [6] U.S. Environmental Protection Agency, <http://www.epa.gov/>.
- [7] Freescale Semiconductor Inc., “Green Embedded Computing and the MPC8536E PowerQUICC® III Processor”, white paper, 2009.
- [8] J. D. Meindl and J. Davis, “The fundamental limit on binary switching energy for tera-scale integration (TSI)”, *IEEE JSSCC*, 35(10), pp. 1515–1516, Oct. 2000.
- [9] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, “The limit of dynamic voltage scaling and insomniac dynamic voltage scaling” *IEEE Transactions on VLSI Systems*, pp. 1239-1252, November 2005.
- [10] Alice Wang, Benton H. Calhoun, and Anantha Chandrakasan. *Sub-threshold Design for Ultra Low-Power Systems*. Springer, 2006.

- [11] A. P. Chandrakasan, D. C. Daly, D. F. Finchelstein, J. Kwong, Y. K. Ramadass, M. E. Sinangil, V. Sze, N. Verma, "Technologies for Ultradynamic Voltage Scaling," Proceedings of the IEEE, vol. 98, no. 2, pp 191-214, February 2010.
- [12] A. Wang and A. Chandrakasan, "A 180mV FFT processor using subthreshold circuit techniques", Digest of Technical Papers, ISSCC 2004, pp. 292–293, San Francisco, Feb. 2004.
- [13] M. Nomani, M. Anis and G. Koley, "Statistical Approach for Yield Optimization for Minimum Energy Operation in Sub-Threshold Circuits Considering Variability Issues", IEEE Transactions on Semiconductor Manufacturing, pp. 77-86, Vol. 23, Issue: 1, February 2010.
- [14] B. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits", IEEE J. Solid-State Circuits, vol. 40, no. 9, pp. 1778–1786, Sep. 2005.
- [15] Benini, L. and Micheli, G. d. Dynamic Power Management: Design Techniques and CAD Tools. Kluwer Academic Publishers, 1998.
- [16] L. Benini, A. Bogliolo, and G. De Micheli, G., "A survey of design techniques for system-level dynamic power management", IEEE Trans. Very Large Scale Integr. Syst., pp. 299–316, Jun., 2000.
- [17] Tajana Simunic, "ENERGY EFFICIENT SYSTEM DESIGN AND UTILIZATION", PhD Dissertation, EE Dept., Stanford University, Feb. 2001.
- [18] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," JSSC, vol. SC-30, pp. 847–854, Aug. 1995.

- [19] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," DAC, pp. 495–500, June 1998.
- [20] Shi-Hao, Chen, and Jiing-Yuan, Lin, "Implementation and Verification Practices of DVFS and Power Gating", International Symposium on VLSI Design, Automation and Test, pp. 19–22, Apr, 2009.
- [21] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design" Proceedings IEEE Symposium on Low Power Electronics, pp. 8–11, October 1994.
- [22] A. P. Charandrasan and R. W. Broderson. Low Power Digital CMOS Design. Norwell, MA: Kluwer Academic, 424 pages, 1995.
- [23] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. D. Micheli, "Dynamic voltage scaling for portable systems", In Proceedings of the Design Automation Conference, pp. 524–529, June 2001.
- [24] Intel XScale Processor, "Intel XScale Core Developer's Manual", <http://download.intel.com/design/intelxscale/27347302.pdf>.
- [25] AMD Athlon processor, "Mobile AMD Athlon 4 processor model 6 cpga data sheet", online: <http://www.amd.com>.
- [26] IBM Power PC, Online: <http://www-03.ibm.com/technology/ges/semiconductor/power/powerpc.html>.
- [27] Intel Pentium Processor Family, online: http://www.intel.com/products/desktop/processors/pentium.htm?iid=prod_desktopcore+tab_pentium.
- [28] David Goodwin, Chris Rowen, and Grant Martin, "Configurable Multi-Processor Platforms for Next Generation Embedded Systems", ASP-DAC, pp. 744-746, 2007.

- [29] Tim Kogel, Heinrich Meyr, “Heterogeneous MP-SoC: the solution to energy-efficient signal processing”, Design Automation Conference, pp. 686- 691, 2004.
- [30] Chris Rowen, Steve Leibson. Engineering the Complex SOC: Fast, Flexible Design with Configurable Processors, Prentice Hall, 2004.
- [31] Federico Angiolini, Jianjiang Ceng, Rainer Leupers, Federico Ferrari, Cesare Ferri, Luca Benini, “An integrated open framework for heterogeneous MPSoC design space exploration”, Design, Automation and Test in Europe conference, pp. 1145-1150, 2006.
- [32] Tensilica Xtensa technology, available online: www.tensilica.com/products/xtensa_LX.htm.
- [33] Sungdae Choi, Seong-Jun Song, Kyomin Sohn, Hyejung Kim, Jooyoung Kim, Namjun Cho, Jeong-Ho Woo, Jerald Yoo and Hoi-Jun Yoo, “A Multi-Nodes Human Body Communication Sensor Network Control Processor”, IEEE Custom Integrated Circuits Conferences, pp. 109–112, Sep., 2006.
- [34] Nazhandali, L., Minuth, M., Zhai, B., Olson, J., Austin, T., and Blaauw, “A second-generation sensor network processor with application-driven memory optimizations and out-of-order execution” In Proceedings of the 2005 international Conference on Compilers, Architectures and Synthesis For Embedded Systems, San Francisco, California, USA, pp. 249–256, September 24 - 27, 2005.
- [35] L. Nazhandali, “Architectural Optimization for Performance- and Energy-Constrained Sensor Processors”, PhD thesis, EECS Dept., University of Michigan, 2006.
- [36] Ekanayake, V., Kelly, C., and Manohar, R., “An ultra low-power processor for sensor networks”, In Proceedings of the 11th international Conference on Architectural

Support For Programming Languages and Operating Systems, Boston, MA, USA, pp. 27–36, October 07 - 13, 2004.

[37] Ahmed Amine Jerraya, Olivier Franza, Markus Levy, Masao Nakaya, Pierre G. Paulin, Ulrich Ramacher, Deepu Talla, Wayne Wolf, “Envisioning the Future for Multiprocessor SoC”, IEEE Design & Test of Computers, pp. 174–183, June 2007.

[38] Ahmed Jerraya and Wayne Wolf. Multiprocessor Systems-on-Chips. Elsevier, Sep, 2004.

[39] Sébastien Le Beux, Guy Bois, Gabriela Nicolescu, Youcef Bouchebaba, Michel Langevin, Pierre G. Paulin, “Combining mapping and partitioning exploration for NoC-based embedded systems”, Journal of Systems Architecture - Embedded Systems Design, pp. 223-232, July 2010.

[40] Mohammad Hossein Neishaburi, Zeljko Zilic, “Reliability aware NoC router architecture using input channel buffer sharing”, ACM Great Lakes Symposium on VLSI, pp. 511-516, 2009.

[41] S. Borkar, “Design Challenges of Technology Scaling”, IEEE Micro, July, 1999.

[42] P., Zarkesh-Ha, J. A. Davis and J. D. Meindl, ”Prediction of net-length distribution for global interconnects in a heterogeneous system-on-a-chip”, IEEE Trans. Very Large Scale Integr. Syst., pp. 649-659, Dec., 2000.

[43] N. Magen, A. Kolodny, U. Weiser and N. Shamir, “Interconnect-power dissipation in a microprocessor“, In Proceedings of the international Workshop on System Level interconnect Prediction, pp. 7-13, Feb., 2004.

[44] H. B. Bakoglu and J. D. Meindl, “Optimal interconnection circuits for VLSI”, IEEE Trans. Electron Devices, vol. ED-32, pp. 903–909, 1985.

- [45] A. Nalamalpu and W. Burleson, "A practical approach to DSM repeater insertion: Satisfying delay constraints while minimizing area and power", in IEEE ASIC/SOC Conference, pp. 152 - 156, Sep., 2001.
- [46] Chen, G. and Friedman, E. G., "Low-power repeaters driving RC and RLC interconnects with delay and bandwidth constraints", IEEE TVLSI, pp. 161-172, February, 2006.
- [47] Kaul, H., Sylvester, D., Blaauw, D., Mudge, T., and Austin, T., "DVS for On-Chip Bus Designs Based on Timing Error Correction", pp. 80 – 85, DATE, March, 2005.
- [48] K. Banerjee and A. Mehrotra, "A power-optimal repeater insertion methodology for global interconnects in nanometer designs", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 49, no. 11, pp. 2001-2007, Nov. 2002.
- [49] A. Nalamalpu, S. Srinivasan, and W. P. Burleson, "Boosters for driving long onchip interconnects—Design issues, interconnect synthesis, and comparison with repeaters", IEEE Trans. Comput.-Aided Des., vol. 21, no. 1, pp. 50–62, Jan. 2002.
- [50] C. Alpert, A. Devgan, and S. Quay, "Buffer insertion for noise and delay optimization", IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 18, no. 11, pp. 1633-1645, Nov. 1999.
- [51] L. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal elmore delay", in Proc. Int. Symp. Circuits Syst. (ISCAS), 1990, pp. 865-868.
- [52] V. Adler and E. G. Friedman, "Repeater design to reduce delay and power in resistive interconnect", IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 45, no. 5, pp. 607-616, May 1998.

- [53] Yehea I. Ismail , Eby G. Friedman, “Effects of inductance on the propagation delay and repeater insertion in VLSI circuits”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, v.8 n.2, p.195-206, April 2000.
- [54] Akl, C. J. and Bayoumi, M. A., “Reducing interconnect delay uncertainty via hybrid polarity repeater insertion”, IEEE Trans. Very Large Scale Integr. Syst., pp. 1230-1239, Sep., 2008.
- [55] Pontes, J., Moreira, M., Soares, R., and Calazans, N., “Hermes-GLP: A GALS Network on Chip Router with Power Control Techniques”, In Proceedings of the International Symposium on VLSI (ISVLI), pp. 347 - 352, Apr, 2008.
- [56] Semeraro, G., Magklis, G., Balasubramonian, R., Albonesi, D. H., Dwarkadas, S., and Scott, M. L., “Energy-Efficient Processor Design Using Multiple Clock Domains with Dynamic Voltage and Frequency Scaling”, HPCA, pp. 29-40, 2002.
- [57] Chen, G., Li, F., Kandemir, M., and Irwin, M., “Reducing NoC energy consumption through compiler-directed channel voltage scaling”, SIGPLAN Not. 41, 6, pp. 193-203, Jun. 2006.
- [58] Li, F., Chen, G., Kandemir, M., and Kolcu, I., “Profile-driven energy reduction in network-on-chips”, SIGPLAN Not. 42, 6, pp. 394-404, Jun. 2007.
- [59] Cheung, N., Parameswarani, S., and Henkel, J., “A quantitative study and estimation models for extensible instructions in embedded processors”, In Proceedings of the IEEE/ACM international Conference on Computer-Aided Design, pp. 183-189, November, 2004.

- [60] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations", Proceedings of Int. Symposium on Computer Architecture, pp. 340-345, June 2000.
- [61] J. Russel and M. Jacome, "Software power estimation and optimization for high performance 32-bit embedded processors", Proceedings of Int. Conf. Computer Design, pp. 328-333, Oct, 1998.
- [62] S. Chandra, K. Lahiri, A. Raghunathan, and S. Dey, "Variation-Aware System-Level Power Analysis", IEEE Trans. on VLSI, pp. 1173-1184, Aug., 2010.
- [63] A. P. Chandrakasan and A. Sinha, "JouleTrack: A Web Based Tool for Software Energy Profiling", Proceedings of Design Automation Conference, pp. 220-225, June, 2001.
- [64] Y. Fei, S. Ravi, A. Raghunathan, and N. K. Jha, "A Hybrid Energy-Estimation Technique for Extensible Processors", IEEE Trans. on CAD, vol. 23, pp. 652-664, May, 2004.
- [65] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy" in Proceedings IEEE Annual Foundations Computer Science, pp. 374-382, 1995.
- [66] I. Hong, G. Qu, M. Potkonjak, and M. B. Srivastava, "Synthesis techniques for low-power hard real-time systems on variable-voltage processors", Proc. Realtime Systems Symposium, pp. 178--187, 1998.
- [67] G. Quan and X. Hu, "Minimum energy fixed-priority scheduling for variable voltage processors", in Proc. Design Automation Test Euope, pp. 782-787, Mar 2002.

- [68] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors", in Proceedings International Symp. Low-Power Electron Design, pp. 197–202, 1999.
- [69] G. Dhiman, and T. Simunic Rosing, "System level power management using online learning", IEEE Trans. on CAD, pp. 676-689, May, 2009.
- [70] K. Choi, R. Soma, and M. Pedram, "Fine-grained dynamic voltage and frequency scaling for precise energy and performance trade-off based on the ratio of off-chip access to on-chip computation times", IEEE Trans. on CAD, Vol. 24, No. 1, pp.18-28, Jan 2005.
- [71] S. Lee and T. Sakurai, "Run-time power control scheme using software feedback loop for low-power real-time applications", in Proc. Asia-Pacific Design Automation Conference, pp. 381–386, 2000.
- [72] A. Azevedo, I. Issenin, R. Cornea, R. Gupta, N. Dutt, A. Veidenbaum, and A. Nicolau, "Profile-based dynamic voltage scheduling using program checkpoints in the COPPER framework", in Proc. Design Automation Test Eur. Conference, pp. 168–176, Mar 2002.
- [73] C. Hsu and U. Kremer, "Compiler-Directed Dynamic Voltage Scaling for Memory-Bound Applications", Dept. Comput. Sci., Rutgers Univ., New Brunswick, NJ, Tech. Rep. DCS-TR-498, Aug. 2002.
- [74] E.-Y. Chung, L. Benini, and G. De Micheli, "Contents provider-assisted dynamic voltage scaling for low energy multimedia applications", in Proc. IEEE Int. Symp. Low-Power Design, Monterey, CA, pp. 42–47, Aug. 2002.

- [75] P. Yang, C. Wong, P. Marchal, F. Catthoor, D. Desmet, D. Verkest, and R. Lauwereins, "Energy-aware runtime scheduling for embedded multiprocessor SoCs", *IEEE Design and Test Computers*, vol. 18, no. 5, pp. 46–58, Sep. 2001.
- [76] J. Cong and K. Gururaj, "Energy Efficient Multiprocessor Task Scheduling under Input-dependent Variation", *Proceedings of Design, Automation and Test in Europe*, pp. 411-416, April 2009.
- [77] David C. Snowdon, Etienne Le, Sueur Stefan, M. Petters, and Gernot Heiser, "Koala: A Platform for OS-Level Power Management", *Eurosys*, pp. 289-302, Apr, 2009.
- [78] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities", *AFIPS Conference*, pp. 483–485, 1967.
- [79] Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "Estimation of energy performance in computing platforms", *IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, pp. 723-726, Dec., 2009.
- [80] X. Feng, R. Ge, and K. Cameron, "Power and energy profiling of scientific applications on distributed systems", *Proc. 19th Int'l Parallel & Distributed Processing Symp. (IPDPS 05)*, pp. 4-8, Apr. 2005.
- [81] Lahiri, K., Dey, S., Panigrahi, D., and Raghunathan, A., "Battery-Driven System Design: A New Frontier in Low Power Design", In *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 261-267, January, 2002.
- [82] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J., "Wireless sensor networks for habitat monitoring", In *Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia, USA, pp. 88-97, September 28 - 28, 2002.

- [83] Eduardo Casilari, Jose M. Cano-García and Gonzalo Campos-Garrido, “Modeling of Current Consumption in 802.15.4/ZigBee Sensor Motes”, pp. 5443-5468, *Sensors*, June, 2010.
- [84] Pablo Suarez, Carl-Gustav Renmarker, Adam Dunkels, and Thiemo Voigt, “Increasing ZigBee network lifetime with X-MAC”, In Proceedings of the workshop on Real-world wireless sensor networks, pp. 26-30, 2008.
- [85] Liu Yu, Zhang Wei, and K. Akkaya, “Static worst-case energy and lifetime estimation of wireless sensor networks”, International Performance Computing and Communications Conference (IPCCC), pp. 17-24, Dec., 2009.
- [86] M. Tariq, M. Macuha, Park Yong-Jin, and T. Sato, “An Energy Estimation Model for Mobile Sensor Networks”, International Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 507-512, July, 2010.
- [87] P. Gburzynski, B. Kaminska, and W. Olesinski, “A Tiny and Efficient Wireless Ad-hoc Protocol for Low-cost Sensor Networks”, pp. 1562-1567, DATE 2007.
- [88] O.S. Unsal and I. Koren, "System-level power-aware design techniques in real-time systems", Proceedings of the IEEE, vol.91, no.7, pp. 1055- 1069, July 2003.
- [89] G. W. Miao, N. Himayat, G. Y. Li, and A. Swami, “Cross-layer optimization for energy-efficient wireless communications: a survey” (invited), *Wiley Journal Wireless Commun. and Mobile Computing*, vol.9, no.4, pp. 529-542, Apr. 2009.
- [90] Dongliang Duan, Fengzhong Qu, Liuqing Yang, Ananthram Swami, and Jose C. Principe, "Modulation Selection from a Battery Power Efficiency Perspective", *IEEE Transactions on Communications*, vol.58, no.7, pp.1907-1911, July 2010.

- [91] T. Sakurai, "Approximation of Wiring Delay in MOS-FET LSI", *IEEE Journal of Solid-State Circuits*, vol. 4, pp. 418-426, Aug., 1983.
- [92] J. Rabaey, A. Chandrakasan and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. 2nd edition, Prentice Hall, 2003.
- [93] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *J. Applied Phys.*, vol. 19, no.1, pp. 55–63, Jan. 1948.
- [94] Cong, J. and Pan, D. Z., "Wire width planning for interconnect performance optimization", *IEEE Trans. Comput. Aided Des. Integrated Circuits*, pp. 319-329, Mar, 2002.
- [95] Houman Zarrabi, Haydar Saaied, A.J.Al-Khalili and Yvon Savaria, "Zero Skew Differential Clock Distribution Network", *International Symposium on Circuit And Systems (ISCAS)*, pp. 2077-2080, May, 2006.
- [96] J. Zhang and E. G. Friedman, "Decoupling Technique and Crosstalk Analysis of Coupled *RLC* Interconnects," *Proceedings of the IEEE International Symposium on Circuits and Systems*, Vol. II, pp. 521-524, May 2004.
- [97] A. B. Kahng, S. Muddu and E. Sarto, "On switch factor based analysis of coupled RC interconnects", In *DAC*, pp. 79-84, June, 2000.
- [98] L. Pileggi, "Coping with RC(L) Interconnect Design Headaches", *IEEE ICCAD Tutorial*, pp. 246-253, Nov. 1995.
- [99] G. Yee, R. Chandra, V. Ganesan and C. Sechen, "Wire Delay in the Presence of Crosstalk", *ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pp. 170-175, Dec. 1997.

- [100] C. Duan, B. J. LaMeris and S. Khatri. On and Off-chip Cross-talk Avoidance in VLSI Designs. Springer, 2010.
- [101] Zhao, W. and Cao, Y., “New Generation of Predictive Technology Model for Sub-45nm Design Exploration”, IEEE International Symposium on Quality Electronic Design (ISQED), pp. 585-590, 2006.
- [102] Berkeley Predictive Technology Model [Online]. Available: <http://www-device.eecs.berkeley.edu/~ptm>.
- [103] X. C. Li, J. F. Mao, H. F. Huang, and Y. Liu, “Global interconnect width and spacing optimization for latency, bandwidth and power dissipation” IEEE Trans. Elec. Devices, vol. 52, pp. 2272-2279, October 2005.
- [104] Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, “An Interconnect-Aware Delay Model for Dynamic Voltage Scaling in nm Technologies”, GLSVLSI, 2009.
- [105] L. Macchiarulo, E. Macii, M. Poncino. “Low-Energy Encoding for Deep-Submicron Address Buses”, ISLPED, pp. 176-181, 2001.
- [106] Kalyan, T. V., Mutyam, M., and Rao, P. V., “Exploiting Variable Cycle Transmission for Energy-Efficient On-Chip Interconnect Design”, International Conference on VLSI Design, pp. 235-241, 2008.
- [107] Ghoneima, M., Ismail, Y., Khellah, M. M., Tschanz, J., and De, V., “Serial-link bus: a low-power on-chip bus architecture”, Trans. Cir. Sys. Part I, pp. 2020 - 2032, Sep., 2009.
- [108] Magklis, G., Scott, M. L., Semeraro, G., Albonesi, D. H., and Dropsho, S., “Profile-Based Dynamic Voltage and Frequency Scaling for a Multiple Clock Domain

Microprocessor, International Symposium on Computer Architecture (ISCA), pp. 14-27, 2003.

[109] V. Kursun and E. G. Friedman. *Multi-Voltage CMOS Circuit Design*, West Sussex, England, John Wiley & Sons Press, 2006.

[110] Sinha, A., A. P. Chandrakasan, "Dynamic Power Management in Wireless Sensor Networks", *IEEE Design and Test*, pp.62-74, April, 2001.

[111] P. P. Sotiriadis and A. P. Chandrakasan, "A bus energy model for deep submicron technology", *IEEE Trans. Very Large Scale Integr. Syst.*, pp. 341-350, June, 2002.

[112] Sotiriadis, P., A. P. Chandrakasan, "Bus Energy Reduction by Transition Pattern Coding Using a Detailed Deep Submicrometer Bus Model" *IEEE Transactions on Circuits and Systems*, pp. 1280-1295, October 2003.

[113] H. Deogun, R.M. Rao, D. Sylvester, R. Brown, and K. Nowka, "Dynamically pulsed MTCMOS with bus encoding for total power and crosstalk minimization," *IEEE International Symposium on Quality Electronic Design*, pp. 88-93, 2005.

[114] L. Benini, G. De Micheli, E. Macii, D. Sciuto and S. Silvano, "Address bus encoding techniques for system-level power optimization", In *Proceedings of the Conference on Design, Automation and Test in Europe*, February, pp. 861-866, 1998.

[115] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities", *AFIPS Conference*, pp. 483-485, 1967.

[116] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*, 4th Ed., Morgan Kaufmann, 2007.

[117] A. P. Chandrakasan and S. Sheng and R. W. Brodersen, "Low-Power CMOS Digital Design", *JSSC*, V27, N4, pp 473--484, April 1992.

- [118] Xilinx Web Power Tool, available online: http://www.xilinx.com/cgi-bin/power_tool/power_Virtex2.
- [119] N. Beucher, N. Bélanger, Y. Savaria, and G. Bois, “A Methodology to Evaluate the Energy Efficiency of Application Specific Processors”, IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 983-986, 2007.
- [120] Mentor Graphics ModelSim, online: www.mentor.com.
- [121] K. Hilman, H. W. Park, and Y. Kim, “Using Motion-Compensated Frame-Rate Conversion for the Correction of 3:2 Pulldown Artifacts in Video Sequences,”, IEEE TCAS for video technology, vol. 10, no. 6, pp. 869-877, Sept., 2000.
- [122] Beucher, N., Bélanger, N., Savaria, Y., and Bois, G., “High Acceleration for Video Application Using Specialized Instruction Set based on Parallelism and Data Reuse”, Journal of Signal Processing Systems (JSPS), pp. 155-165, 2009.
- [123] Fibonacci Numbers, available online: en.wikipedia.org/wiki/Fibonacci_number.
- [124] C. Christopoulos, A. Skodras, T. Ebrahimi, “The JPEG2000 still images coding system: an overview”, IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp. 1103-1127, Nov., 2000.
- [125] Lam, S., Shoaib, M., and Srikanthan, T, “Modeling Arbitrator Delay-Area Dependencies in Customizable Instruction Set Processors”, in proceedings of the third IEEE international workshop on electronic Design, Test and Applications, pp. 237-242, January 17 - 19, 2006).
- [126] Paolo Ienne and Rainer Leupers, Customizable Embedded Processors-Design Technologies and Applications. Systems on Silicon Series, Morgan Kaufmann, San Mateo, California, 2006.

- [127] Frequency. Available: <http://en.wikipedia.org/wiki/Frequency>.
- [128] Texas Instrument ZigBee[®] compliant protocol stack. Available: www.ti.com/z-stack.
- [129] Texas Instrument Sensor Demo Software. Available: <http://www.ti.com/litv/zip/swrc147b>.
- [130] Texas Instrument Second Generation System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 / RF4CE / ZigBee. Available: <http://focus.ti.com/docs/prod/folders/print/cc2530.html>.
- [131] Houman Zarrabi, A. J. Al-Khalili and Yvon Savaria, "An Interconnect-Aware Dynamic Voltage Scaling Scheme for DSM VLSI", In Proceedings of International Symposium on Circuit and Systems (ISCAS), Paris, France, pp. 41-44, May, 2010.
- [132] —, "Repeater Insertion in Power-Managed VLSI", to appear in Great Lakes Symposium on VLSI (GLSVLSI), Lausanne, Switzerland, May, 2011.
- [133] —, "Modeling, Design and Management of Interconnects in Power-Managed VLSI", submitted to Transactions on Circuits and Systems.
- [134] —, "Design Space Exploration of Interconnect Repeaters in Power-Managed VLSI", submitted to Journal on Emerging and Selected Topics in Circuits and Systems.
- [135] —, "Estimation of Energy Performance in Computing Platforms", submitted to Transactions on VLSI.
- [136] —, "Activity Management in Battery-Powered Embedded Systems: A Case Study of ZigBee[®] WSN", submitted to VLSI-SoC, Oct, 2011.